

Marco José Pereira Tomás

Automatização de uma máquina de corte de pedra de
3 eixos

Tese de Mestrado

Engenharia Eletrotécnica - Energia e Automação Industrial

Professor Doutor Miguel Lima



RESUMO

A indústria é o grande motor da economia e só com a automatização industrial se consegue competir nos mercados internacionais. A indústria transformadora de rochas continua em crescimento e, num contexto regional, existem dezenas de instalações industriais em funcionamento há algumas décadas. Destas unidades, algumas têm vindo a investir em novos equipamentos e na remodelação de equipamentos mais antigos.

Neste trabalho desenvolveu-se um sistema de automação que permite modernizar máquinas de corte de pedra de três eixos, acrescentando a possibilidade de corte através do desenho de um perfil de corte e controlo de custos. Este sistema controla a máquina e interage com o operador. O interface homem máquina é simples, de modo a permitir uma fácil adaptação dos operadores sem necessidade de mão-de-obra especializada.

Em termos práticos, esta modernização consiste em refazer o quadro elétrico, substituindo os sistemas de controlo a relé ou autómatos antigos, por um novo autómato associado a variadores de velocidade e a um computador. O computador executa uma aplicação de interface homem máquina amigável que permite dar ordens de controlo à máquina, bem como definir parâmetros, introduzir medidas e desenhar as retas ou arcos que constituem os perfis de corte.

ABSTRACT

Industry is the biggest engine of the economy, only with industrial automation can compete in international markets. Stone industry continues to grow and in a regional context there are some industrial units laboring for few decades. Some of these units have been investing in new equipment and refurbishment of older equipment.

With this work we intend to develop an automation system that allows the modernization of three axes stone cutting machinery adding features as cutting through the drawing and cost control. This system will control the machine and interact with the operator. The human-machine interface is simple and easy to use without the need of specialized workers.

In practical terms, this modernization consists in replacing the relay control systems or old programmable logic controller by a new programmable logic controller associated with variable speed drives and a computer. The computer runs a friendly human-machine interface program to control the machine, set parameters, introduce measures and draw lines or arcs that constitute the cutting profiles.

PALAVRAS CHAVE

Automação industrial
Autômato programável
PLC
Interface homem máquina
HMI
Sistema de controlo
Atuadores
Sensores
Supervisão
CNC
Motores assíncronos
Variadores de frequência
Codificador
OPC
Visual Basic

KEY WORDS

Industrial Automation
Programmable Controllers
PLC
Human Machine Interface
HMI
Control system
Actuators
Sensors
Supervision
CNC
Asynchronous Motors
Frequency Inverters
Encoder
OPC
Visual Basic

ÍNDICE GERAL

ÍNDICE GERAL	ix
ÍNDICE DE FIGURAS	xi
ÍNDICE DE TABELAS	xiii
ABREVIATURAS E SIGLAS	xv
1. Introdução	1
1.1 Contexto	1
1.2 Objetivos	1
1.3 Estrutura da dissertação	2
2. Sistemas de controlo utilizados em máquinas	3
2.1 Introdução aos sistemas de controlo	3
2.2 Autómatos programáveis (PLCs)	4
2.3 Programação de PLCs VIPA	6
2.3.1 Tipos de Blocos de Programa	6
2.3.2 Estrutura do Programa	7
2.3.3 Imagens de Processo	7
2.3.4 Execução cíclica do programa	7
2.4 Interface homem máquina (HMI)	10
2.5 Controlo numérico computadorizado (CNC)	11
2.6 Sistemas adotados frequentemente por fabricantes de máquinas de corte de pedra ..	13
3. Sistema de controlo de máquina de corte de pedra baseado em PLC e HMI	15
3.1 Requisitos gerais do sistema de controlo	15
3.2 <i>Hardware</i>	17
3.2.1 Características dos motores da máquina	17
3.2.2 Variadores de frequência	18
3.2.3 Sensores indutivos	22
3.2.4 Codificadores	23
3.2.5 Sensor de fluxo de água	28
3.2.6 Pressostato	29

3.2.7	Contador de energia	29
3.2.8	Autômato programável	30
3.2.9	HMI.....	37
3.3	<i>Software</i>	38
3.3.1	<i>Software</i> do autômato.....	38
3.3.2	<i>Software</i> HMI.....	45
3.3.3	Interface de comunicação entre o PC e o PLC.....	55
3.4	Análise económica	56
4.	Resultados	59
5.	Conclusões e desenvolvimentos futuros	63
	Referências.....	65
	Anexo 1	67
	Esquema elétrico de potência e comando da máquina.....	67
	Anexo 2	97
	Código do OPC	97
	Código da função desenhar	102
	Código da função determinar centro de circunferência	103
	Código da função de cálculo da coordenada Z	104
	Código da função de cálculo dos pontos de corte do molde	104

ÍNDICE DE FIGURAS

Figura 1: Máquina de corte de pedra	2
Figura 2: Sistema de controlo	3
Figura 3: Exemplo de PLC	4
Figura 4: Execução cíclica de programa.....	9
Figura 5: Exemplo de HMI.....	10
Figura 6: Componentes das máquinas de controlo numérico.....	12
Figura 7: Sistema de controlo de máquina de corte de pedra.....	15
Figura 8: Motor assíncrono trifásico	17
Figura 9: Variador de frequência do fabricante KEB.....	18
Figura 10: Sensor indutivo	22
Figura 11: Esquema da saída do sensor indutivo PNP	23
Figura 12: Codificador rotativo	24
Figura 13: Impulsos na saída de um codificador incremental	25
Figura 14: Disco codificado de 8 bits.....	25
Figura 15: Codificador ótico.....	27
Figura 16: Sensor de fluxo IFM SI5010.....	28
Figura 17: Montagem do sensor de fluxo IFM SI5010	28
Figura 18: Pressostato diferencial	29
Figura 19: Contador de energia	29
Figura 20: Configuração de <i>hardware</i> do autómato utilizado.....	30
Figura 21: CPU (VIPA 214-1BA02).....	31
Figura 22: Módulo de entradas digitais (VIPA 221-1BH30)	32
Figura 23: Módulo de saídas digitais (VIPA 222-1BH30).....	33
Figura 24: Módulo de saídas digitais a relé (VIPA 222-1HF00)	34
Figura 25: Módulo de entradas analógicas (VIPA 231-1BD53)	35
Figura 26: Módulo de saídas analógicas (VIPA 232-1BD51).....	36
Figura 27: Módulo de contagem (VIPA 250-1BA00).....	37
Figura 28: Fluxograma do programa do bloco OB1.....	40
Figura 29: Limites de corte no plano YZ	42
Figura 30: Fluxograma da função Corte FC9	43
Figura 31: Ecrã inicial	46
Figura 32: Ecrã Desenhar	47
Figura 33: Ecrã Corte de Chapa	48
Figura 34: Ecrã Cortar por Molde	49
Figura 35: Fluxograma da abertura de ficheiro de molde	50
Figura 36: Fluxograma da rotina Desenhar	51
Figura 37: Fluxograma da rotina Calcular Molde	53

Figura 38: Ecrã de Dados.....	54
Figura 39: Arquitetura de um sistema OPC	55
Figura 40: Máquina de corte de pedra	59
Figura 41: Ecrã Corte de Chapa.....	60
Figura 42: Máquina a efetuar corte de chapa.....	61
Figura 43: Ecrã Cortar por Molde.....	61
Figura 44: Máquina a efetuar a maquinação da peça pretendida.....	62
Figura 45: Peça maquinada	62

ÍNDICE DE TABELAS

Tabela 1: Interface de controlo do variador KEB F5 de 1,5kW	19
Tabela 2: Interface de controlo do variador KEB F5 de 22kW	21
Tabela 3: Conversão binário natural – código de Gray	26
Tabela 4: Entradas e saídas digitais do PLC.....	38
Tabela 5: Custo dos equipamentos	56

ABREVIATURAS E SIGLAS

CAD	Desenho Assistido por Computador (<i>Computer Aided Design</i>)
CAM	Manufatura Assistida por Computador (<i>Computer Aided Manufacturing</i>)
CNC	Controlo Numérico Computorizado (<i>Computer Numerical Control</i>)
CPU	Unidade Central de Processamento (<i>Central Processing Unit</i>)
DC	Corrente Contínua (<i>Direct Current</i>)
EMC	Compatibilidade Eletromagnética (<i>Electromagnetic Compatibility</i>)
GUI	Interface Gráfico da Aplicação (<i>Graphical User Interface</i>)
HMI	Interface Homem Máquina (<i>Human Machine Interface</i>)
IDE	Ambiente de Desenvolvimento Integrado (<i>Integrated Development Environment</i>)
NC	Controlo Numérico (<i>Numerical Control</i>)
OPC	<i>OLE for Process Control</i>
PID	Controlador Proporcional Integral Derivativo (<i>Proportional Integral Derivative</i>)
PLC	Autómato Programável (<i>Programmable Logic Controller</i>)

1. Introdução

1.1 Contexto

A pedra natural portuguesa tem características únicas, sendo este setor responsável por 1,5% das exportações do país, posicionando Portugal entre os primeiros dez produtores e exportadores de Pedra Natural a nível mundial [1].

Numa altura em que as exportações são cruciais para a economia, este sector aposta na internacionalização através da modernização, da inovação e da melhoria da competitividade.

No âmbito da modernização surgiu a necessidade deste projeto, pois nesta região do país existem várias unidades fabris de transformação de rochas naturais, com algumas dezenas de anos e com alguns equipamentos obsoletos.

Dentro da gama de equipamentos mais vulgares nesta indústria encontram-se as máquinas de corte a disco de 3 eixos, pelo que este projeto consiste no desenvolvimento de um sistema de automação que permita a modernização de máquinas deste tipo.

1.2 Objetivos

As máquinas de corte de pedra com disco utilizam discos diamantados. O processo de corte é realizado através da rotação do disco e movimento dos eixos. Estas máquinas possibilitam o corte de pedras planas e pedras moldadas¹, como se mostra na Figura 1.

¹ Pedras cortadas através de um perfil de corte elaborado com retas ou arcos

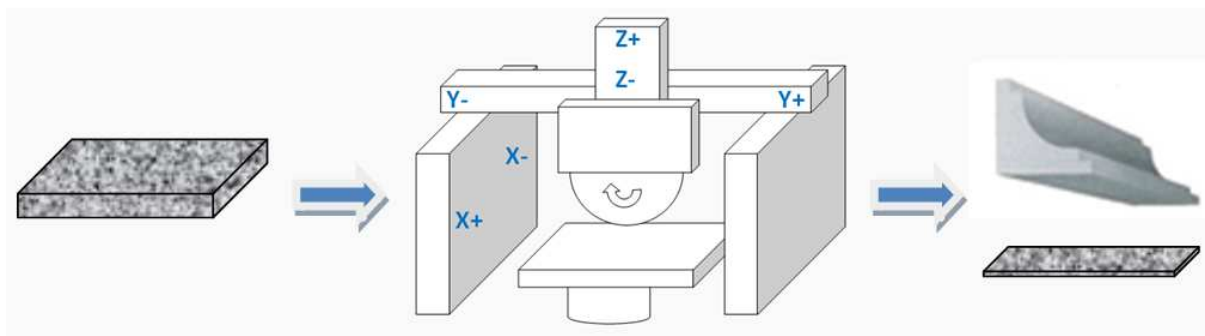


Figura 1: Máquina de corte de pedra

Do ponto de vista do utilizador, este trabalho tem como objetivo principal o desenvolvimento de um *software* de interface homem máquina HMI² residente num computador que permita introduzir medidas, desenhos de moldes e dar ordens de controlo à máquina.

De forma a implementar as funções referidas, é necessário dotar um autómato de um conjunto de primitivas que possibilitem o controlo da máquina e o interface de comunicação entre o computador e o autómato. Do ponto de vista elétrico o sistema será constituído por uma parte de comando e por uma parte de potência.

Este projeto pode-se aplicar, com pequenos ajustes, a uma grande variedade de máquinas de corte de pedra existentes na indústria.

1.3 Estrutura da dissertação

O presente documento encontra-se organizado em cinco capítulos e dois anexos. Seguidamente serão descritos sucintamente os conteúdos presentes em cada capítulo e anexos. Neste capítulo de introdução apresentam-se os objetivos do trabalho e faz-se o enquadramento do problema.

O capítulo 2 apresenta a tecnologia normalmente utilizada nos sistemas de controlo de máquinas com movimento de eixos.

No capítulo 3 propõe-se uma solução para controlar a máquina de corte de pedra com a análise do *hardware* elétrico da máquina e análise dos programas desenvolvidos.

No capítulo 4 apresentam-se os resultados práticos deste projeto.

Finalmente, no capítulo 5 apresentam-se as principais conclusões que decorrem do trabalho realizado. Neste capítulo referem-se ainda alguns aspetos de evolução futura do trabalho.

Para além dos capítulos referidos existem dois anexos.

O anexo 1 apresenta o esquema elétrico de comando e de potência de toda a máquina.

No anexo 2 mostram-se alguns excertos de código do programa em Visual Basic que pretendem ilustrar alguns aspetos de programação.

² Interface Homem Máquina

2. Sistemas de controlo utilizados em máquinas

Na automatização de máquinas, a eletrónica digital e os microprocessadores vieram substituir os antigos sistemas de controlo baseados em sistemas eletromecânicos. Apareceram assim os autómatos programáveis, os interfaces homem máquina gráficos e o controlo numérico computadorizado. Neste capítulo apresenta-se a tecnologia normalmente utilizada em equipamentos similares ao desenvolvido neste trabalho.

2.1 Introdução aos sistemas de controlo

Para que um sistema de automação industrial, de qualquer natureza, funcione de forma correta e segura é necessário implementar um sistema de controlo apropriado. Independentemente da dimensão ou da complexidade do processo em causa, o respetivo sistema de controlo pode ser decomposto em três subsistemas com funções bem definidas: o operador humano, o controlador e o processo controlado como se mostra na Figura 2 [2].

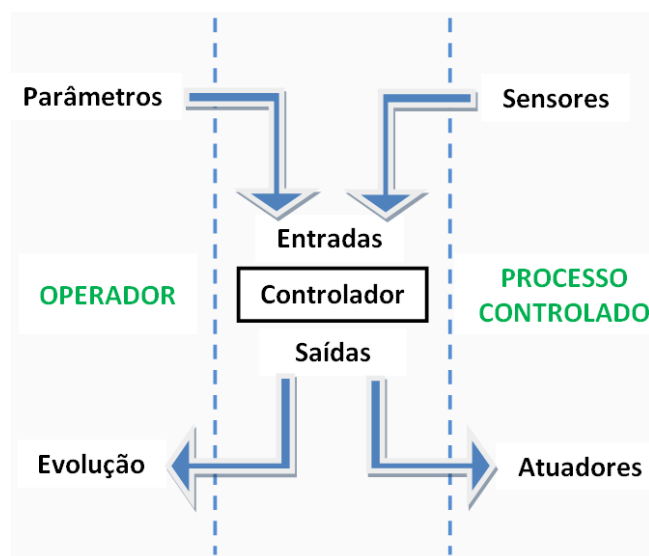


Figura 2: Sistema de controlo

O controlador assume uma grande importância, uma vez que é responsável pela execução do algoritmo de controlo adequado ao funcionamento correto do processo a controlar. Este interage com o ambiente que o rodeia através de duas interfaces com diferentes características. A interface com o processo controlado é definida como a interface de instrumentação e a interface com o operador humano é a interface homem-máquina.

A interface de instrumentação é composta por um conjunto de sensores e atuadores que, através de um conjunto de sinais, permitem ao controlador, não só obter informações sobre o estado do sistema, mas também permitem-lhe alterar o estado do sistema através da manipulação de atuadores.

A interação com o operador humano faz-se através da interface homem-máquina, consistindo num conjunto de dispositivos de entrada e saída que permitem que aquele tenha acesso ao estado do processo e execute algumas funções de controlo. Tipicamente, esta interação realiza-se ao nível da definição de parâmetros do processo e da supervisão da respetiva evolução.

Como já referido, o controlador executa o algoritmo adequado de modo a condicionar a evolução do processo. Através do processamento da informação obtida, quer diretamente do estado do processo, através da interface de instrumentação, quer fornecida pelo operador humano, através da interface homem máquina, o algoritmo de controlo produz um conjunto de comandos que são enviados para o processo através da interface de instrumentação.

2.2 Autómatos programáveis (PLCs)

Desde o final da década de setenta, os PLCs substituíram as soluções até aí adotadas, compostas por instalações cabladas que usavam relés. Os PLCs combinam o *software* e módulos eletrónicos de entrada e saída.

Um PLC (Figura 3) é um sistema baseado em microprocessador, que controla equipamentos industriais e processos. Ele foi inicialmente concebido para realizar funções executadas por relés, interruptores, temporizadores e contadores mecânicos [3].



Figura 3: Exemplo de PLC

Os PLCs modernos possuem funcionalidades de cálculo avançadas. Adicionalmente, as potencialidades dos PLCs podem ser aumentadas com a introdução de módulos de entradas e saídas analógicas, módulos de comunicação, módulos de contagem, entre outros.

Os PLCs são equipamentos fiáveis, versáteis e flexíveis, podendo ser o seu *hardware* e *software* facilmente modificados para novos requisitos de controlo.

Em termos funcionais os PLCs têm como principal característica examinar o estado das entradas, de forma a controlar e atuar sobre as suas saídas. A implementação só é possível, devido à execução de um programa armazenado em memória do programa, na qual o PLC monitoriza continuamente o estado do sistema, através de sinais originados nos dispositivos de entrada.

Tradicionalmente os PLCs têm quatro etapas sequenciais de operação, que são repetidas, no estado atual da tecnologia, milhares de vezes por segundo:

- Auto-Teste (*Self Test*) – Verifica se todos os módulos estão sem erros, faz *reset* ao *watch-dog timer*, que é um temporizador utilizado para prevenir que erros de *software* impeçam o PLC de executar as etapas normais de operação;
- Leitura de entradas – Lê os valores físicos dos módulos de entradas e copia-os para a memória;
- Execução do programa – Baseando-se nos valores em memória das entradas, executa o programa atualizando em memória os valores das saídas;
- Atualização das saídas – Copia os valores das saídas em memória para os módulos de saídas.

Existem diversos fabricantes de PLCs. Os principais fabricantes situam-se nos EUA, Europa e Japão. Após uma análise de custo/benefício optou-se pela escolha do PLC do fabricante VIPA, por este produzir PLCs compatíveis em *hardware* e *software* com o reconhecido fabricante Siemens, mas com um custo mais reduzido.

2.3 Programação de PLCs VIPA

Os PLCs da VIPA são programáveis com a aplicação Step7 da Siemens ou com a aplicação WinPLC 7 da VIPA.

Estes PLCs oferecem vários tipos de blocos, nos quais o programa de utilizador e os seus dados podem ser armazenados. Dependendo das necessidades do processo, o programa pode estar estruturado em diferentes blocos.

2.3.1 Tipos de Blocos de Programa

Bloco de Organização (OB)

Os Blocos de Organização (OBs) formam o corpo principal do programa do utilizador. O programa inteiro pode ser armazenado no OB1 (programa linear), ou pode ser dividido e armazenado em vários blocos (programa estruturado). Os OBs são chamados ciclicamente pelo sistema operativo.

Função FC, SFC

Uma função (FC) é uma rotina que contém uma parte funcional do programa. É possível programar funções de modo que sejam parametrizáveis. Com isso as funções são ideais para serem reutilizadas no programa e para realizarem tarefas repetidas como pode ser o caso, por exemplo, de cálculos complexos.

As funções de sistema (SFC) são funções parametrizáveis integradas no sistema operativo, sendo o seu número e funcionalidade fixos [5].

Bloco de Função FB, SFB

Basicamente os blocos de função oferecem as mesmas possibilidades que as funções. Adicionalmente, os blocos de função possuem a sua própria área de memória (*instance data blocks*). Com isso as funções são ideais para serem reutilizadas no programa e para realizarem tarefas complexas como o controlo em malha fechada.

Por outro lado, os blocos de Funções de Sistema (SFB), são funções parametrizáveis integradas no sistema operativo. De forma semelhante às SFCs, o seu número e funcionalidade são fixos.

Blocos de Dados

Os Blocos de Dados (DB) são áreas de dados do programa do utilizador, nas quais os respetivos dados são geridos de maneira estruturada.

2.3.2 Estrutura do Programa

Os programas dos PLCs podem ser estruturados de diversas formas que se apresentam seguidamente:

Programa Linear

O programa completo encontra-se num bloco contínuo (normalmente OB1).

Este modelo assemelha-se ao de um controlo feito com relés e que foi substituído por um controlador lógico programável. O CPU³ processa as instruções individuais de forma consecutiva.

Programa Particionado

O programa está dividido em blocos, dentro dos quais cada bloco contém somente um programa destinado a solucionar uma tarefa parcial. O bloco de organização OB1 contém instruções para a chamada de outros blocos numa sequência definida.

Programa Estruturado

Um programa estruturado contém blocos com parâmetros ou parametrizáveis. Estes blocos são criados de forma a serem reutilizados.

Ao realizar a chamada de um bloco parametrizável são fornecidos parâmetros (os endereços de entradas e saídas, assim como valores de parâmetros).

2.3.3 Imagens de Processo

Durante o processamento de um programa, o CPU acede a áreas de memória onde os dados binários dos módulos de entrada e saída são armazenados:

PII – tabela de imagem de processo de entrada

PIQ – tabela de imagem de processo de saída

2.3.4 Execução cíclica do programa

O programa desenvolvido pelo utilizador é constituído funcionalmente pelo programa principal que é executado ciclicamente. No entanto, existe também a necessidade de efetuar a inicialização de variáveis, atender interrupções, controlar as comunicações e executar subrotinas, sendo estas tarefas executadas de forma não cíclica.

³ Unidade Central de Processamento

Inicialização

O CPU executa um *restart* completo ao chamar o OB100, ou quando desligado/ligado o interruptor de STOP --> RUN. Durante um *restart* completo, o sistema operativo apaga os bits de memória não-retentivos, os temporizadores, os contadores, a pilha de interrupção e a pilha de blocos. Faz-se um *reset* em todas as interrupções de *hardware* armazenadas e interrupções de diagnóstico e inicia a monitorização do tempo de ciclo de *scan* [5].

Ciclo de Scan

A operação cíclica do CPU consiste em três tarefas principais, de acordo com o diagrama da Figura 4:

- O CPU verifica o estado dos sinais de entrada e atualiza a memória imagem das entradas;
- Executa as instruções do programa do utilizador;
- Escreve os valores provenientes da memória imagem das saídas nos módulos de saída.

Para além das três tarefas principais referidas, existe um procedimento que permite verificar o tempo de processamento de cada ciclo. Este procedimento faz também *reset* ao *watch-dog timer*.

Existe também um conjunto de tarefas complementares, tais como o Auto-teste e a comunicação com outros equipamentos, nomeadamente outros PLCs, consolas HMI e ambientes de programação.

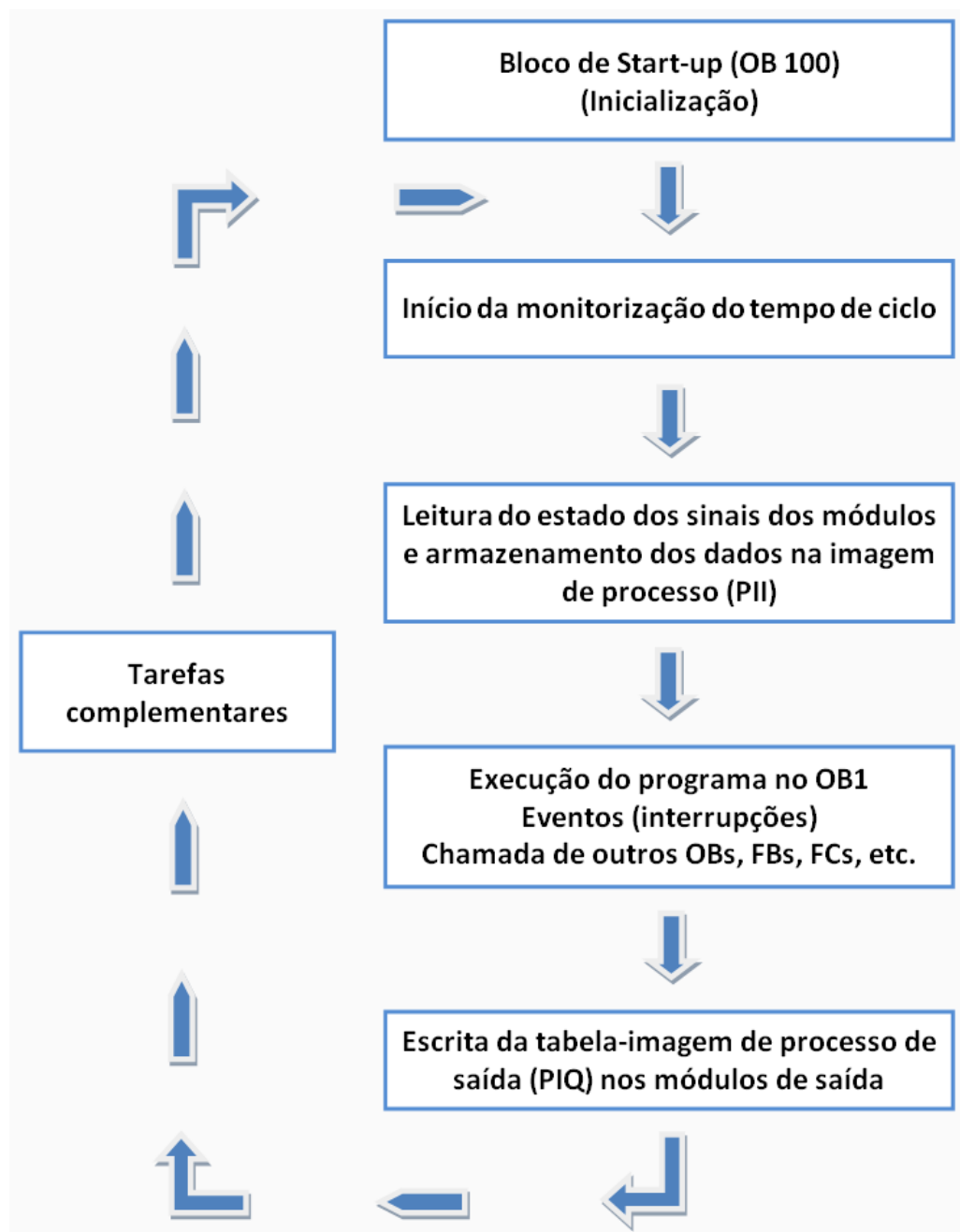


Figura 4: Execução cíclica de programa

2.4 Interface homem máquina (HMI)

O objetivo básico de um HMI⁴ (Figura 5) é fornecer ao operador um interface que permita facilmente controlar e monitorizar o processo.

Os HMI são frequentemente usados para:

- Mostrar alarmes;
- Mostrar o estado da máquina;
- Possibilitar ao operador iniciar e parar ciclos;
- Possibilitar a introdução de valores.

Vantagens:

- Menor quantidade de botões físicos e cablagens para controlar o processo;
- Fácil reconhecimento do estado do processo através de desenhos ilustrativos do processo;
- Possibilidade de acesso a diferentes níveis de informação mudando de ecrãs.



Figura 5: Exemplo de HMI

⁴ Interface Homem Máquina

2.5 Controlo numérico computadorizado (CNC)

O controlo numérico é uma forma de automação programável, no qual o equipamento de processamento é controlado por programas que permitem executar sequências de operações complexas dentro da máquina, sem o auxílio do operador humano.

Uma máquina NC⁵ é, em termos funcionais, semelhante a uma máquina convencional, diferindo na maneira como as funções e os movimentos da máquina são controlados.

Os componentes típicos de uma máquina de comando numérico são:

- Eixos de deslocamento;
- Transmissões;
- Dispositivos de medida de posição e de deslocamento;
- Ferramenta principal ou cabeça (árvore);
- Sistemas de aperto das peças;
- Sistemas de mudança das ferramentas;
- Eixos complementares de rotação e de deslocamento.

Além das funções geométricas para o controlo dos deslocamentos, os sistemas CNC⁶ dispõem de outras funções para a gestão da máquina como, por exemplo, controlo da velocidade da árvore, posicionamento angular da cabeça, utilização do refrigerante de corte, garantia da velocidade de avanço e de corte constantes e a mudança de ferramenta.

O número de funções e a forma como estas se executam dependem da própria máquina de comando numérico e das possibilidades da unidade de controlo.

Componentes de um sistema CNC

O componente principal de um sistema CNC é um computador que se encarrega de realizar todos os cálculos necessários, bem como as operações lógicas de controlo.

Atendendo a que o sistema CNC é a ponte entre o operador e a máquina-ferramenta, são necessárias as respetivas interfaces representadas na Figura 6:

- A interface do operador, formada pelo painel de controlo e vários dispositivos a ele ligados, geralmente relacionados com o armazenamento de programas;
- A interface de controlo da máquina-ferramenta que está subdividida em múltiplas conexões de controlo e que afetam os atuadores dos eixos, da árvore principal e equipamentos auxiliares [6].

⁵ Controlo Numérico

⁶ Controlo Numérico Computorizado

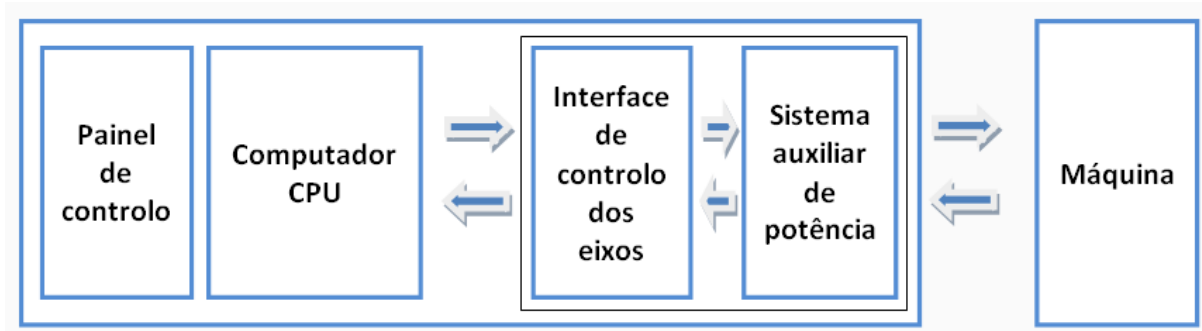


Figura 6: Componentes das máquinas de controlo numérico

Os sistemas CNC apresentam várias vantagens relativamente aos sistemas convencionais. No entanto, sobretudo por se tratar de uma tecnologia bastante específica, apresenta também algumas desvantagens como se refere seguidamente.

Vantagens do CNC:

- Maior flexibilidade, porque estas máquinas permitem realizar um maior número de operações comparativamente com os sistemas convencionais;
- Aumento da produtividade das máquinas devido à diminuição do tempo total de maquinagem, devido à diminuição dos tempos de deslocamento e à rapidez dos posicionamentos possibilitados pelos sistemas eletrónicos de controlo;
- Maior precisão porque, geralmente, os elementos mecânicos e os seus mecanismos beneficiam de maior desenvolvimento tecnológico, conduzindo à maior precisão das máquina ferramenta CNC;
- Alteração do programa da peça de modo fácil e rápido;
- Programação de peças com maior complexidade;
- Facilidade em guardar os programas no controlador ou em registos magnéticos;
- Facilidade de comunicação com um computador;
- Permite que apenas um operador controle mais que uma máquina;
- Possibilidade de se integrar num sistema de Fabrico Assistido por Computador.

Desvantagens do CNC:

- Investimento inicial elevado;
- Manutenção exigente e especializada;
- Necessita de operadores mais especializados;
- Necessita de elevadas taxas de utilização para viabilizar o investimento.

2.6 Sistemas adotados frequentemente por fabricantes de máquinas de corte de pedra

Quando se projeta uma máquina, tem de se adotar o sistema de controlo mais apropriado à execução das funções pretendidas. Para máquinas de corte simples, a maioria dos fabricantes utiliza como sistema de controlo um PLC e uma consola gráfica de interface homem máquina. Em máquinas com mais funcionalidades utilizam-se sistemas CNC.

As primeiras máquinas de corte por molde utilizavam como referência um molde de madeira, em que o sistema de controlo utilizava um sensor de distância para ler o molde e reproduzia-o na pedra. Este sistema não era muito prático porque tinham de se elaborar em madeira os moldes pretendidos.

Mais tarde surgiram máquinas em que as coordenadas de corte do molde são introduzidas na consola gráfica de interface. Deste modo o operador tem a tarefa trabalhosa de determinar todas as coordenadas de corte.

Em máquinas CNC podem ser executados trabalhos de maior complexidade recorrendo a programas de CAD⁷/CAM⁸. A operação destas máquinas exige mão-de-obra mais especializada.

Com este projeto pretende-se elaborar um sistema de controlo baseado num PLC e num interface HMI de simples utilização. O operador, utilizando poucos comandos, criará facilmente os perfis de corte que pretende maquinar, com possibilidades de os guardar e alterar facilmente.

⁷ Desenho Assistido por Computador

⁸ Manufatura Assistida por Computador

3. Sistema de controlo de máquina de corte de pedra baseado em PLC e HMI

Neste capítulo, tendo como referência a estrutura mecânica de uma máquina de corte de pedra de três eixos, desenvolve-se todo o projeto de automatização da máquina.

Começa-se com a análise dos principais equipamentos necessários e disponíveis para a concretização deste projeto. Após reunidas as informações relativas aos equipamentos elétricos foi elaborado o esquema elétrico da máquina, disponível no anexo 1.

Posteriormente tendo em consideração o esquema elétrico da máquina e as funcionalidades pretendidas é desenvolvido o *software* do autómato e *software* de interface homem-máquina.

Por fim, faz-se uma breve análise económica da solução proposta.

3.1 Requisitos gerais do sistema de controlo

O sistema de controlo da máquina, representado na Figura 7, deverá em função das informações dos sensores e das instruções do operador, controlar os atuadores de modo a realizar as tarefas pretendidas.

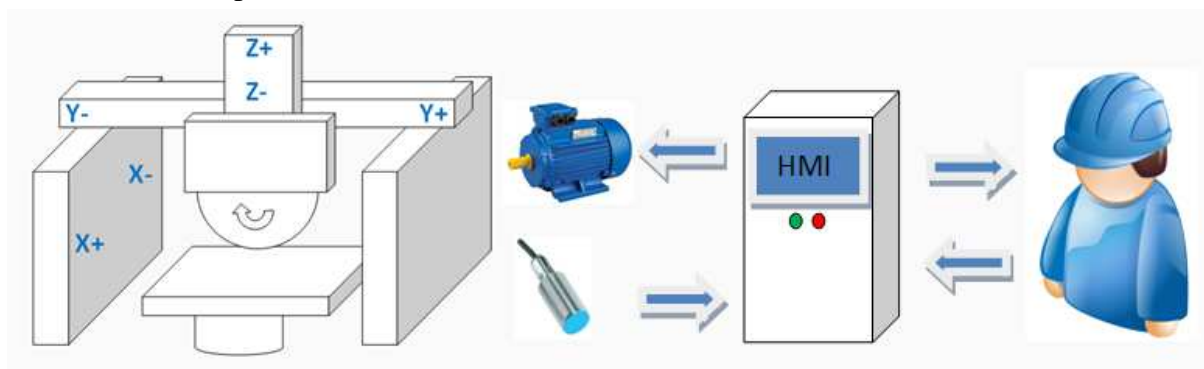


Figura 7: Sistema de controlo de máquina de corte de pedra

Associado ao sistema de controlo para cada eixo X, Y e Z têm-se:

- Sensores de fim de curso para indicarem ao controlador os limites físicos da máquina;

- Um codificador⁹ para que o controlador conheça a posição do eixo;
- Um atuador constituído por um motor assíncrono trifásico associado a um variador de velocidade.

Para a cabeça de corte temos igualmente um motor assíncrono trifásico associado a um variador de velocidade, podendo também ter-se um sensor para realimentação da velocidade de rotação do disco.

O controlador deverá também controlar outros equipamentos auxiliares, tais como:

- Sistema de lubrificação;
- Válvula ou bomba de água de arrefecimento da ferramenta de corte (disco);
- Travões de bloqueio dos eixos;
- Laser de alinhamento.

O controlador deverá monitorizar através de sensores outras grandezas, nomeadamente:

- Pressão de água;
- Nível de óleo de lubrificação;
- Estado das porcas dos fusos que existem em algumas máquinas associadas ao sistema mecânico de deslocamento do eixo Z.

Em termos de interação com o operador, o sistema de controlo deverá possibilitar:

- Ligar e desligar o motor de corte;
- Movimentar os eixos de forma manual;
- Abrir e fechar a água de arrefecimento;
- Monitorizar o estado da máquina;
- Alterar parâmetros de configuração;
- Introduzir as medidas das peças que a máquina vai cortar;
- Fazer desenhos de peças por molde;
- Dar ordens de iniciar e parar o ciclo de funcionamento automático.

⁹ Encoder na literatura anglo-saxónica

3.2 *Hardware*

Nesta secção faz-se uma análise dos principais equipamentos necessários e disponíveis para a concretização deste projeto.

Como se pretende remodelar uma máquina, sem alteração das suas características mecânicas e mantendo os motores elétricos existentes, vai-se então reunir a informação sobre os acionamentos mecânicos existentes, para que se possa prosseguir para a análise e seleção de variadores de frequência, sistema de controlo e outros equipamentos.

3.2.1 Características dos motores da máquina

Os motores de movimentação dos eixos e motor da cabeça de corte são motores assíncronos trifásicos (Figura 8) e apresentam as características abaixo descritas.



Figura 8: Motor assíncrono trifásico

Motores de movimentação dos eixos X, Y e Z:

- Tipo: Assíncrono trifásico;
- Potência: 1,5 kW;
- Frequência: 50 Hz;
- N° de pólos: 4;
- Velocidade a plena carga: 1410 RPM;
- Tensão: 230 V / 400 V;
- Corrente: 5,9 A / 3,5 A;
- Binário a plena carga: 10,2 Nm;
- $\cos(\varphi) = 0,82$ a plena carga.

Motor da cabeça de corte:

- Tipo: Assíncrono trifásico;
- Potência: 22 kW;
- Frequência: 50 Hz;

- N° de pólos: 4;
- Velocidade a plena carga: 1460 RPM;
- Tensão: 400 V / 690 V;
- Corrente: 42,2 A / 24,1 A;
- Binário a plena carga: 141 Nm;
- $\cos(\varphi) = 0,85$ a plena carga.

3.2.2 Variadores de frequência

Para controlo de potência dos atuadores dos eixos utilizam-se variadores de frequência KEB da gama F5.C (Figura 9). Estes variadores de frequência para motores assíncronos caracterizam-se pela sua versatilidade em diversas aplicações e facilidade de parametrização e utilização, devido ao seu *software* de programação. As potências da gama F5.C vão de 0,37kW a 90kW, pelo que se vão utilizar variadores de 1,5kW para os eixos e 22kW para a cabeça de corte.



Figura 9: Variador de frequência do fabricante KEB

As suas principais características técnicas são:

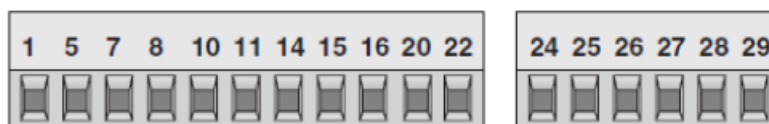
- Entradas digitais programáveis tipo PNP;
- Saídas por relé programáveis;
- Saídas por transístor programáveis;
- Entradas analógicas $\pm 10\text{VDC}$, 0-20mA ou 4-20mA;

- Saídas analógicas $\pm 10\text{VDC}$ programáveis;
- Alimentação 230VAC e 400VAC;
- Módulo de frenagem incorporado de série;
- Filtro EMC¹⁰ classe B incorporado de série;
- Frenagem DC¹¹;
- Controlo PID¹² com ponto de regulação fixo ou variável;
- Frequência de saída até 1600Hz;
- Tempo de ciclo inferior a 2ms.

Interface de controlo do variador KEB F5 de 1,5kW

Na Tabela 1 apresenta-se a interface de controlo do variador, com as especificações dos sinais elétricos em causa.

Tabela 1: Interface de controlo do variador KEB F5 de 1,5kW
X2A



PIN	Function	Name	Description	
Function				
1	± Set value input 1	AN1+	Voltage input 0...±10 VDC ^ 0...±CP.11	Resolution: 11bit scan time: 2ms
Analog output				Imax: 5mA
5	Analog output 1	ANOUT1	Analog output of the output frequency 0...±10 VDC ^ 0...±100 Hz	Ri: 100 Ω Resolution: 12bit
Voltage supply				
7	+10 V Output	CRF	Reference voltage for setpoint potentiometer	+10VDC +5% / max. 4 mA
8	Analog Mass	COM	Mass for analog in- and outputs	
Digital inputs				
10	Fixed frequency 1	I1	I1+I2 = Fixed frequency 3 (default: 70 Hz)	13...30VDC ±0% stabilized Ri=2,1kΩ Scan time 2ms
11	Fixed frequency 2	I2	no input = analog set value	
14	Forward	F	Rotation selection; forward has priority	
15	Reverse	R		
16	Control release / Reset	ST	Power modules are enabled; Error Reset at opening	
Voltage supply				
20	24 V-Output	Uout	Approx. 24V output (max.100 mA))	
22	Digital Mass	0V	Potential for digital in-/outputs	
Relay Outputs				
24	NO contact 1	RLA	Fault relay (default); Function can be changed with CP.31	at maximum 30VDC 0.01...1A
25	NC contact 1	RLB		
26	Switching contact 1	RLC		
27	NO contact 2	FLA	Frequency dependent switch (pre-setting); Function can be changed with CP.32	
28	NC contact 2	FLB		
29	Switching contact 2	FLC		

¹⁰ Compatibilidade Eletromagnética

¹¹ Corrente Contínua

¹² Controlo Proporcional Integral Derivativo

Para este projeto são utilizadas as seguintes entradas e saídas:

- Entrada analógica AN1+ para controlo da frequência;
- Entradas digitais F e R para controlo da direção e ordem de ligar/desligar;
- Entradas digitais I1 e I2 para ativar frequências fixas;
- Entrada digital ST para reset;
- Saída a relé 1 para indicação de falha.






























Em termos de parametrização do variador, mantêm-se os parâmetros de fábrica, excepto os seguintes:

- CP10 – Frequência mínima = 5Hz;
- CP12 – Tempo de aceleração = 1s;
- CP13 – Tempo de desaceleração = 0,1s;
- CP19 – Frequência fixa 1 = 5Hz;
- CP20 – Frequência fixa 2 = 30Hz;
- CP21 – Frequência fixa 3 = 70Hz.

Interface de controlo do variador KEB F5 de 22kW

Na Tabela 2 apresenta-se a interface de controlo do variador, com as especificações dos sinais elétricos em causa.

Tabela 2: Interface de controlo do variador KEB F5 de 22kW

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
																												

PIN	Function	Name	Description
Analog inputs			
1	+ Set value input 1	AN1+	Difference voltage
2	- Set value input 1	AN1-	0...±10 VDC ^ 0...±CP.11
3	+ Analog input 2	AN2+	Input has no function in the CP mode
4	- Analog input 2	AN1-	
Analog outputs			
5	Analog output 1	ANOUT1	Analog output of the output frequency 0...±10VDC ^ 0...±100 Hz
6	Analog output 2	ANOUT2	Analog output of the apparent current 0...10VDC ^ 0...2x rated current
Voltage supply			
7	+10 V Output	CRF	Reference voltage for setpoint potentiometer
8	Analog Mass	COM	Mass for analog in- and outputs
9			
Digital inputs			
10	Fixed frequency 1	I1	I1+I2 = Fixed frequency 3 (default: 70Hz) no input = analog set value
11	Fixed frequency 2	I2	
12	External fault	I3	Input for external fault stopping mode
13	DC braking	I4	Activates the dc braking
14	Forward	F	Rotation selection; forward has priority
15	Reverse	R	
16	Control release / Reset	ST	Power modules are enabled; Error Reset at opening
17	reset	RST	Reset; only when an error occurs
Transistor outputs			
18	Speed dependent	O1	Transistor output switched at actual speed = set speed
19	Ready signal	O2	Transistor output switched, as long as no error occurs
Voltage supply			
20	24 V-Output	Uout	Approx. 24V output (max.100 mA))
21	20...30V-Input	Uin	Voltage input for external supply
22	Digital Mass	0V	Potential for digital in-/outputs
23			
Relay Outputs			
24	NO contact 1	RLA	Fault relay (default); Function can be changed with CP.33
25	NC contact 1	RLB	
26	Switching contact 1	RLC	Frequency dependent switch (pre-setting); Function can be changed with CP.34
27	NO contact 2	FLA	
28	NC contact 2	FLB	
29	Switching contact 2	FLC	

Resolution 12Bit (11 Bit at housing size B)	
Scan time 1 ms	

5 mA; Ri=100 Ω	
Resolution 12Bit	
PWM frequency 3,4 kHz	
Limiting frequency	
Filter 1. Harmonic 178 Hz	

+10VDC +5% / max. 4 mA	
------------------------	--

13...30 VDC ±0% stabilized	
Ri=2,1 kΩ	
Scan time 1 ms	

at maximum 30 VDC	
0.01...1A	

Para este projeto são utilizadas as seguintes entradas e saídas:

- Entrada analógica AN1+ para controlo da frequência;
- Entrada digital F para ligar e desligar o motor;
- Entrada digital ST para *reset*;
- Saída a relé 1 para indicação de falha;

- Saída a relé 2 para indicação de sobrecarga;
- Saída digital O1 para indicação do estado do variador;
- Saída analógica ANOUT2 para visualização da corrente.

Em termos de parametrização do variador, mantêm-se os parâmetros de fábrica excepto os seguintes:

CP10 – Frequência mínima = 5Hz;

CP32 – Função do relé 2 = 7 para o caso de sobrecarga.

3.2.3 Sensores indutivos

Para indicar os limites físicos de início e fim de cada eixo são utilizados sensores indutivos, como os representados na Figura 10, pois são muito comuns na indústria, tendo custo reduzido e elevada fiabilidade. O seu funcionamento baseia-se nas alterações provocadas ao circuito oscilante de uma bobina quando materiais metálicos são aproximados ao campo eletromagnético por ela gerado.

Um sensor de proximidade indutivo consiste numa bobina enrolada em torno de um núcleo de ferrite na cabeça do sensor. Aplicando uma alta frequência a esta bobina é gerado um campo eletromagnético oscilante em torno dela. Este campo é monitorizado por um circuito interno e, quando um objeto metálico se aproxima do campo, são criadas correntes elétricas no objeto (correntes de Foucault). Estas correntes causam um efeito semelhante ao transformador originando a redução da amplitude das oscilações. O circuito de monitorização deteta estas alterações e, em seguida, muda o estado da saída.

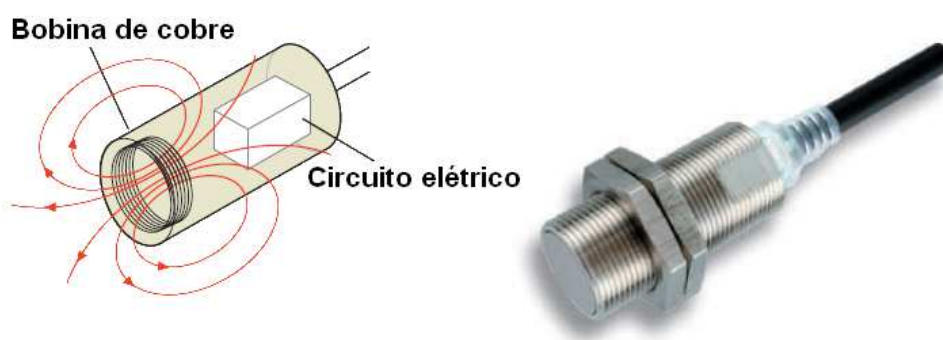


Figura 10: Sensor indutivo

Especificações do sensor E2A-M18KS08-WP-B1 2M do fabricante Omron:

- Diâmetro M18;
- Pré cablado;
- Saída de coletor aberto PNP, representado na Figura 11;
- Distância de deteção 8 mm \pm 10%;

- Deslocamento diferencial máximo 10% da distância de detecção;
- Objeto alvo de metal ferroso (a distância de detecção diminui com metal não ferroso);
- Frequência de resposta de 500 Hz;
- Tensão de alimentação de 12 a 24 VDC;
- Consumo de corrente máximo de 10 mA;
- Corrente de carga máxima de 200 mA (máximo 32 VDC);
- Indicador de funcionamento (LED amarelo);
- Proteção em caso de polarização inversa de saída;
- Proteção em caso de polarização inversa do circuito da fonte de alimentação;
- Dispositivo de supressão de sobretensão;
- Proteção contra curto-circuitos.

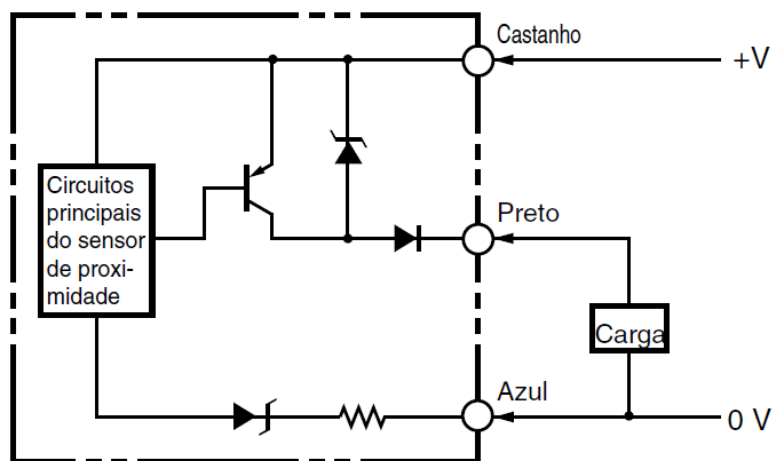


Figura 11: Esquema da saída do sensor indutivo PNP

3.2.4 Codificadores

O componente mais utilizado para determinar a posição dos eixos de uma máquina é o codificador.

Um codificador, mostrado na Figura 12, é um transdutor que converte um movimento angular ou linear em sinais elétricos, isto é, impulsos digitais. Com a utilização de codificadores podem determinar-se grandezas tais como: velocidade, aceleração, distância, rotação e direção.

Seguidamente mostra-se a classificação normalmente adotada para os codificadores e também se descreve o princípio de funcionamento dos codificadores mais utilizados na indústria.



Figura 12: Codificador rotativo

Classificação dos codificadores

Quanto ao princípio de funcionamento, existem codificadores incrementais e codificadores absolutos.

Quanto ao tipo de movimento convertido, os codificadores podem ser rotativos ou lineares.

Em termos da tecnologia de construção os codificadores podem ser óticos ou magnéticos.

Codificadores incrementais

Nestes codificadores a posição é obtida através de impulsos quadrados transmitidos e acumulados ao longo do tempo. Os impulsos são transmitidos através de dois canais A e B desfasados de 90° tal como mostrado na Figura 13.

Para se ler apenas a posição, pode-se utilizar um dos canais A ou B.

A desfasagem entre os dois canais determina o sentido do movimento. Caso o canal A esteja 90° adiantado em relação ao canal B o sentido será horário e se o canal A estiver atrasado 90° em relação ao canal B o sentido será anti-horário.

Existe um outro canal Z de sincronismo, também chamado de “zero” do codificador, que fornece uma posição de referência, gerando um impulso quadrado a cada volta do codificador.

A resolução do codificador incremental é dada por impulsos por volta, isto é, o codificador gera uma certa quantidade de impulsos digitais por cada volta de 360° (no caso de um codificador rotativo);

Para determinar a resolução basta dividir o número de impulsos por 360°. Por exemplo, um codificador de 1024 impulsos por volta, geraria um impulso a cada 0,35° de rotação.

Normalmente, nos codificadores incrementais são disponibilizados, além dos sinais A, B e Z, também os sinais complementares representados por /A, /B e /Z.

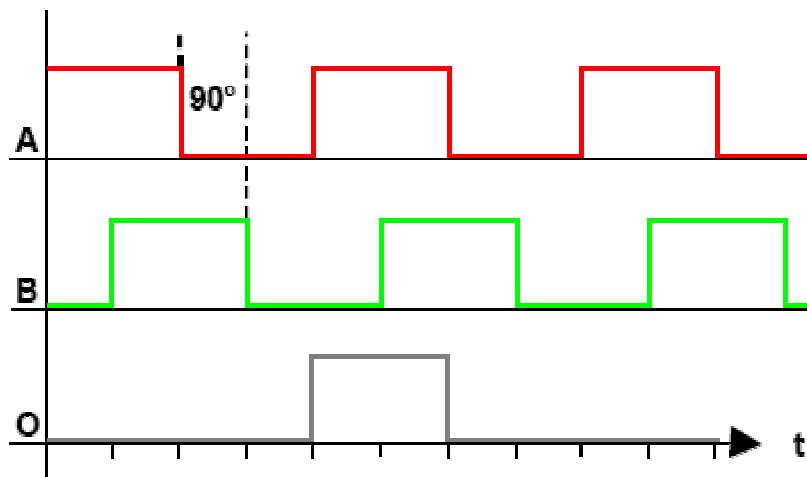


Figura 13: Impulsos na saída de um codificador incremental

Codificadores absolutos

A posição do codificador incremental é dada por impulsos a partir do impulso zero, enquanto a posição do codificador absoluto é determinada pela leitura de um código e este é único para cada posição do seu curso.

Os codificadores absolutos não perdem a posição real no caso de uma eventual falha de energia (até mesmo se deslocados manualmente). Quando voltar a energia ao sistema, a posição é atualizada e disponibilizada na saída (graças ao código gravado no disco do codificador: ver exemplo de disco na Figura 14) e, com isso, não se precisa de ir até a posição zero do codificador para saber a sua localização, como é o caso do incremental.

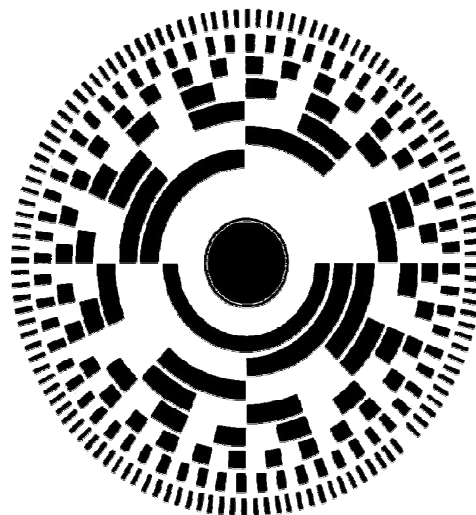


Figura 14: Disco codificado de 8 bits

O código de saída é utilizado para definir a posição absoluta do codificador. O código mais utilizado é o binário, pois este é facilmente manipulado por um circuito relativamente simples

e, com isso, não é necessário nenhum tipo de conversão para se obter a posição real do codificador.

Mas com a utilização internamente de discos codificados em binário natural, o sincronismo e a aquisição da posição no momento da variação entre dois códigos tornam-se muito difíceis. Se nós pegarmos como exemplo dois códigos consecutivos binários como 7 (0111) e 8 (1000), notaremos que a variação de zero para um e um para zero ocorre em todos os bits, e uma leitura feita no momento da transição pode resultar num valor completamente errado.

Para solucionar esse problema é utilizado um código binário chamado "Código Gray", que tem a particularidade de, na comutação de um número para outro consecutivo, somente um bit é alterado como podemos verificar na Tabela 3. Posteriormente (após a leitura do disco), o código Gray pode ser convertido para binário natural, através de um simples circuito de lógica combinacional.

Quando o número de bits da resolução do codificador não é potência de 2, a propriedade de mudar somente um único bit deixa de ser verdadeira no código Gray.

Tabela 3: Conversão binário natural – código de Gray

Decimal	Binário	Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

A resolução do codificador absoluto é dada por contagem por volta, isto é, se ele tiver no seu disco (codificador rotativo) 12 faixas para o código Gray, então terá 2^{12} combinações possíveis, perfazendo um total de 4096 combinações. Fazendo algumas contas, concluiremos que o codificador gera uma combinação de códigos a cada $0,0879^\circ$.

Os codificadores absolutos podem ter a sua resolução definida numa única volta ou em várias voltas. No caso dos codificadores de uma única volta, a cada volta o valor da contagem é

reinicializado; nos codificadores multi-volta o valor de contagem só é reinicializado depois de um certo número de voltas [7].

Codificador rotativo ótico incremental

Os codificadores óticos são constituídos por um ou mais emissores de luz (LED, Néon, ...), um ou mais detetores de luz (foto célula, foto díodo ou foto transístor) e um disco com janelas opacas e transparentes, tal como representado na Figura 15.

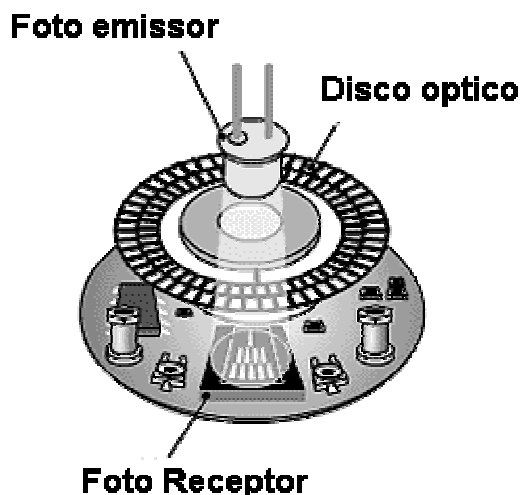


Figura 15: Codificador ótico

As janelas opacas e transparentes têm como função interromper alternadamente o feixe luminoso que, ao incidir no foto detetor, é convertido para impulsos elétricos.

Num codificador incremental, o disco tem as janelas espaçadas uniformemente, alternando as linhas opacas e as transparentes.

Neste projeto, como sensores de posição, utilizam-se codificadores rotativos óticos incrementais, por serem os mais comuns no mercado e por permitirem uma fácil adaptação mecânica a este tipo de máquina. Adicionalmente, este tipo de codificadores apresenta a vantagem de utilizar apenas três fios para envio da informação.

Para os eixos X, Y e Z são utilizados codificadores com as seguintes características:

- 1000 impulsos por volta;
- Frequência máxima de saída 100KHz;
- Velocidade máxima de rotação 2000 voltas/minuto;
- Alimentação 24Vdc;
- Saídas PNP.

3.2.5 Sensor de fluxo de água

Para monitorizar o líquido de arrefecimento da ferramenta de corte utiliza-se um sensor de fluxo de água (Figura 16). Este sensor tem como princípio de funcionamento a deteção da velocidade do fluxo do líquido monitorizado e tem as seguintes características:

- Alimentação 24Vdc;
- Saída PNP 250mA;
- Faixa de medição da velocidade do líquido de 3 a 300 cm/s.

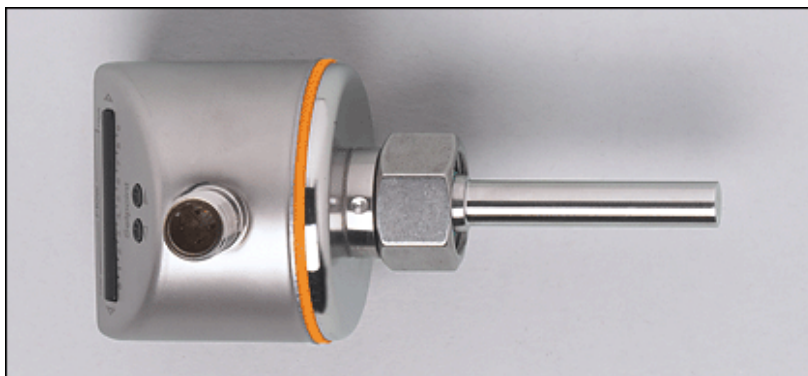


Figura 16: Sensor de fluxo IFM SI5010

A Figura 17 ilustra a montagem do sensor de fluxo IFM SI5010 utilizado neste trabalho. Este sensor deve ser montado no tubo como representado na figura, depois da eletroválvula da água, numa zona onde não haja possibilidade de existir turbulência e deve estar totalmente rodeado pelo líquido que vai monitorizar.

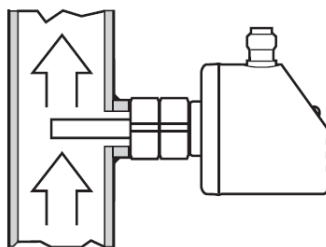


Figura 17: Montagem do sensor de fluxo IFM SI5010

O sensor possui uma escala de leds que permite analisar o seu estado de funcionamento. Usando os dois botões de configuração, ajusta-se o valor de regulação da comutação da saída digital do sensor.

3.2.6 Pressostato

Para monitorização da pressão de ar necessária aos travões de bloqueio da mesa de corte utiliza-se um simples pressostato diferencial, representado na Figura 18, com as características:

- Pressão mínima de 3,5 bar;
- Pressão máxima de 35 bar;
- Saída a relé.



Figura 18: Pressostato diferencial

3.2.7 Contador de energia

De modo a medir os consumos de energia da máquina, utiliza-se um contador de energia com saída por impulsos. O contador Saia, série AWD3, representado na Figura 19, possibilita medições com transformadores de intensidade até 6000A e tem uma saída digital de 10 impulsos por kWh.



Figura 19: Contador de energia

3.2.8 Autómato programável

Como controlador desta máquina utiliza-se um autómato VIPA da série 200. Trata-se de um autómato compacto e modular para aplicações centralizadas ou descentralizadas. Devido à variedade de módulos disponíveis, pode ser utilizado para uma grande diversidade de aplicações.

As principais características deste autómato são:

- Programável com a aplicação Step7 da Siemens;
- Programável com a aplicação WinPLC 7 da VIPA;
- Memória de trabalho integrada, não sendo necessário adicionar cartão de memória;
- Memória ROM integrada para guardar programa e dados;
- Suporta cartão MMC standard;
- Interface MPI incluído (RS232 ou RS485);
- Permite aplicações para gerir áreas de forma centralizada ou descentralizada;
- Possui relógio de tempo real;
- Possibilita a aplicação de até 32 módulos na *rack* do CPU;
- Apresenta um *design* compacto.

Na Figura 20, está representada a configuração de *hardware* do autómato utilizado.

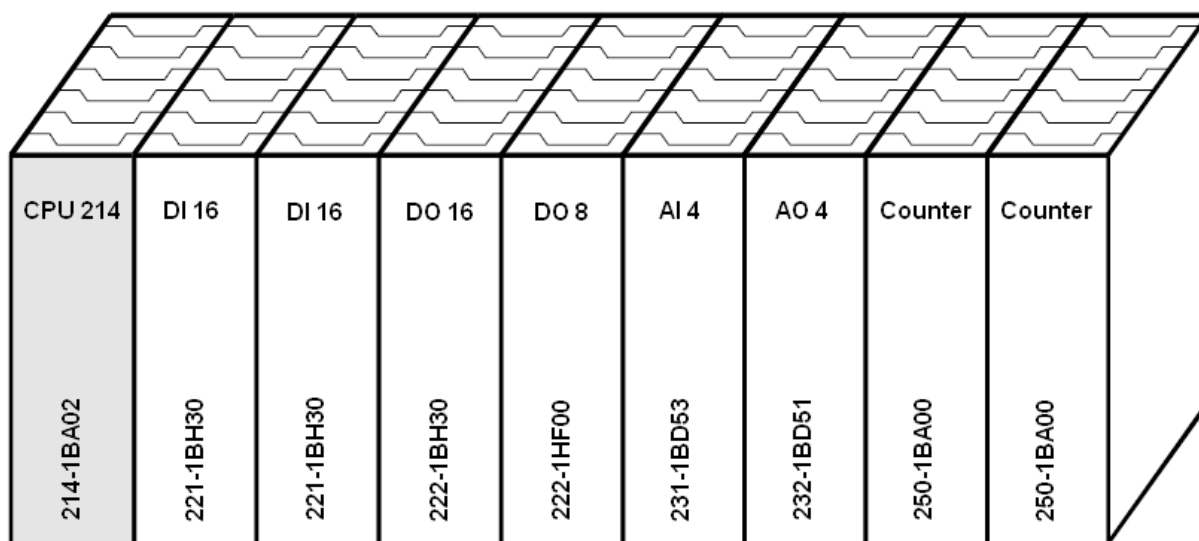


Figura 20: Configuração de *hardware* do autómato utilizado.

Módulo do microprocessador

O módulo do microprocessador utilizado apresenta-se na Figura 21. Trata-se do módulo VIPA 214-1BA02 e as suas características principais são as seguintes:

- Conjunto de instruções compatível com Siemens STEP7;
- Configurável através do Siemens SIMATIC manager;
- Gama de endereços de entrada/saída: 1024Bytes de entradas e 1024Bytes de saídas;
- 48KBytes de memória de trabalho;
- 80KBytes de memória de programa;
- 256 Temporizadores;
- 256 Contadores;
- 8192 Bits de memória;
- Tempo de ciclo em operações de bits de 0,18uS;
- Tempo de ciclo em operações de words de 0,78uS;
- MMC slot;
- Relógio de tempo real;
- Interface MP2I para transferência de dados entre o PC e o CPU;
- Controlador de *Bus* para interligação a periféricos;
- Alimentação de 24 VDC.

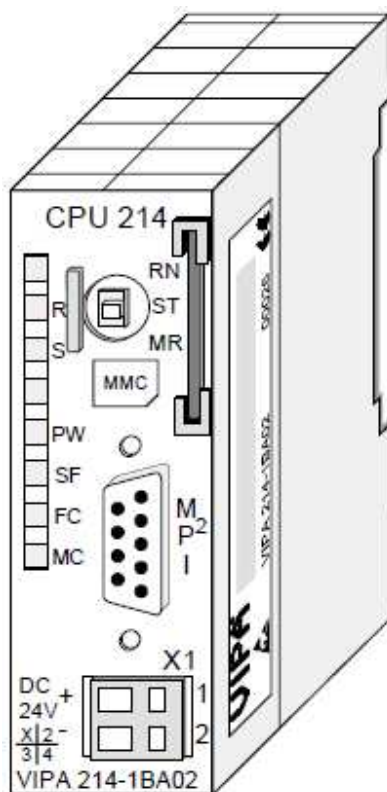


Figura 21: CPU (VIPA 214-1BA02)

Módulo de entradas digitais

O módulo de entradas digitais utilizado apresenta-se na Figura 22. Trata-se do módulo VIPA 221-1BH30 e as suas características principais são as seguintes:

- 16 entradas;
- Tensão nominal: 24 VDC;
- Nível lógico “0”: 0 ... 5 V;
- Nível lógico “1”: 15 ... 28,8V;
- Dados de entrada: 2 Bytes;
- Leds de indicação de estado;
- Alimentação: 5 VDC / 45 mA via Bus.

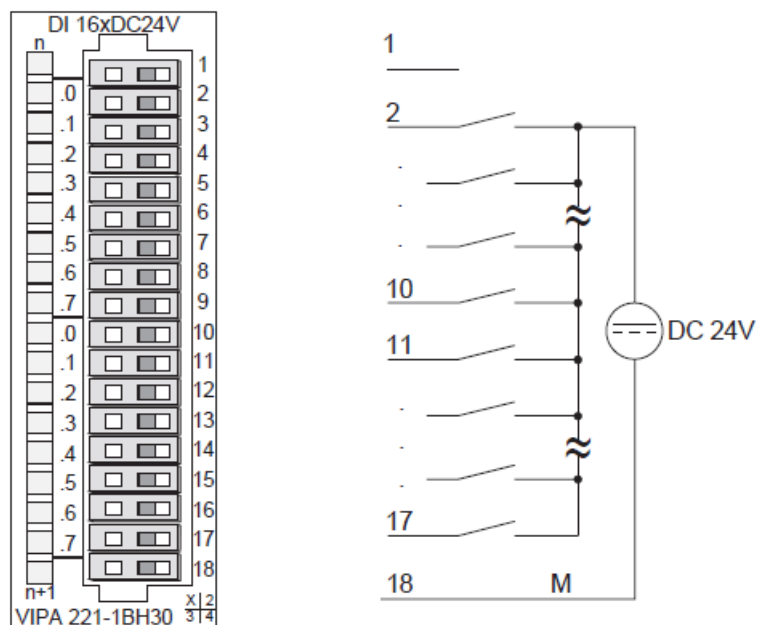


Figura 22: Módulo de entradas digitais (VIPA 221-1BH30)

Módulo de saídas digitais

O módulo de saídas digitais utilizado apresenta-se na Figura 23. Trata-se do módulo VIPA 222-1BH30 e as suas características principais são as seguintes:

- 16 saídas;
- Tensão nominal: 24 VDC;
- Corrente por saída: 0,5A;
- Dados de saída: 2 Bytes;
- Leds de indicação de estado;
- Alimentação: 5 VDC / 120 mA via Bus.

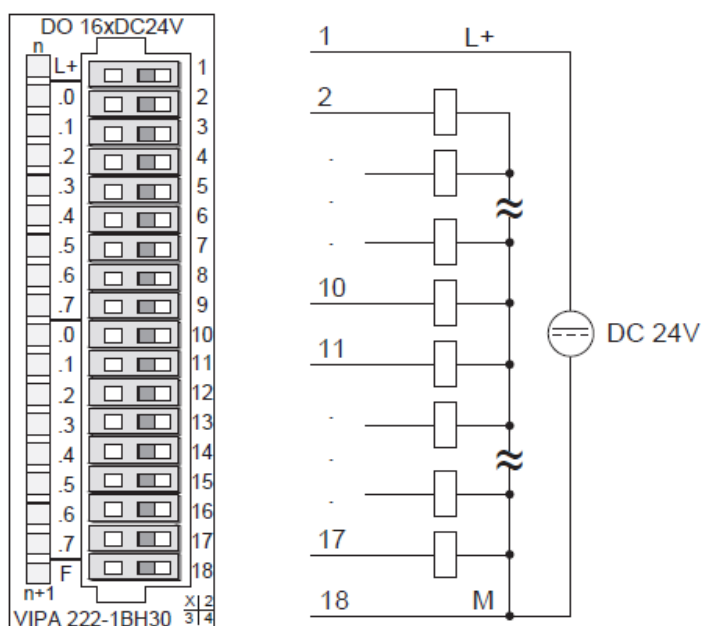


Figura 23: Módulo de saídas digitais (VIPA 222-1BH30)

Módulo de saídas digitais a relé

Outro módulo de saídas digitais utilizado apresenta-se na Figura 24. Trata-se do módulo VIPA 222-1HF00 e as suas características principais são as seguintes:

- 8 saídas a relé;
- Tensão nominal: 30 VDC / 230 VAC;
- Corrente por saída: 5A;
- Dados de saída: 1 Byte;
- Leds de indicação de estado;
- Alimentação: 5 VDC / 300 mA via Bus.

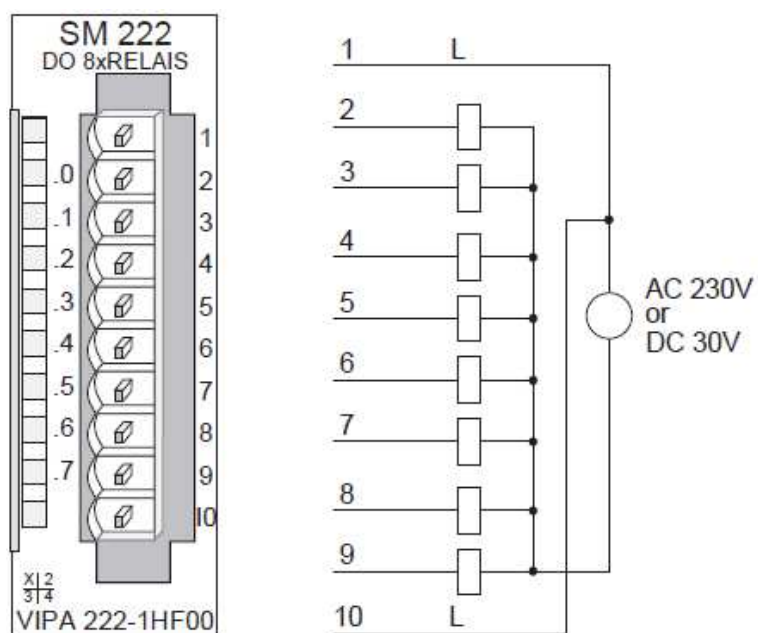


Figura 24: Módulo de saídas digitais a relé (VIPA 222-1HF00)

Módulo de entradas analógicas

O módulo de entradas analógicas utilizado apresenta-se na Figura 25. Trata-se do módulo VIPA 231-1BD53 e as suas características principais são as seguintes:

- 4 entradas com resolução de 16Bit;
- Dados de entrada: 8 Bytes;
- Leds de indicação de estado;
- Alimentação: 5 VDC / 280 mA via Bus;
- As quatro entradas podem ser configuradas individualmente para medição de corrente, tensão ou resistência, permitindo ligar diversos tipo de sensores de temperatura e outros transdutores.

Neste projeto as entradas analógicas são configuradas para medição de tensões de 0 a 10VDC e permitem ao sistema de controlo “saber” a velocidade que o operador pretende, utilizando os potenciómetros de controlo de velocidade dos eixos.

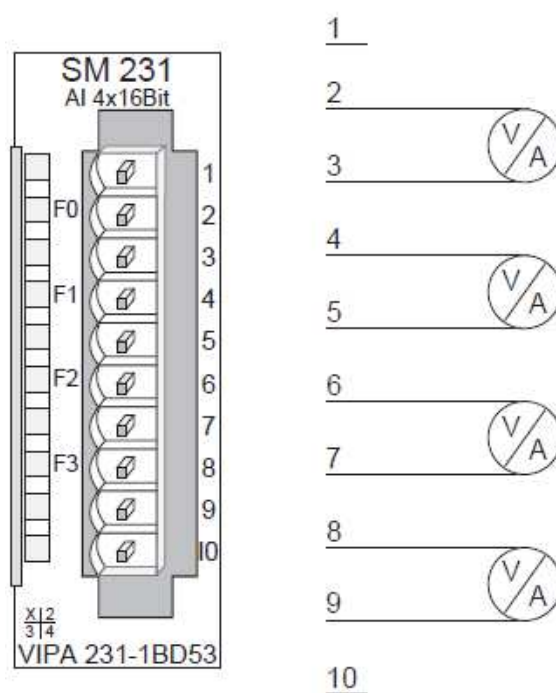


Figura 25: Módulo de entradas analógicas (VIPA 231-1BD53)

Módulo de saídas analógicas

O módulo de saídas analógicas utilizado apresenta-se na Figura 26. Trata-se do módulo VIPA 232-1BD51 e as suas características principais são as seguintes:

- 4 saídas 12Bit;
- Dados de saída: 8 Bytes;
- Leds de indicação de estado;
- Alimentação: 5 VDC / 75 mA via Bus;
- 24 VDC / 60 mA;
- Saídas adequadas para ligação de atuadores que requeiram +-10V, 1 a 5V, 0 a 10V, +-20mA, 4 a 20mA, 0 a 20mA;

Neste projeto as saídas analógicas são configuradas para tensão 0 a 10VDC e permitem ao sistema controlar a velocidade dos motores dos eixos.

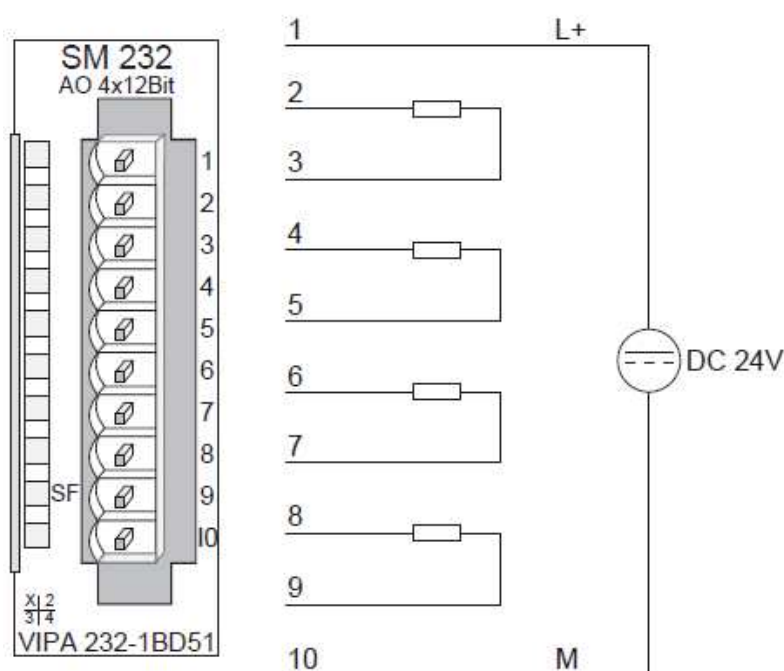


Figura 26: Módulo de saídas analógicas (VIPA 232-1BD51)

Módulo de contagem

O módulo de contagem utilizado apresenta-se na Figura 27. Trata-se do módulo VIPA 250-1BA00 e as suas características principais são as seguintes:

- 2 canais de 32Bit ou 4 canais de 16Bit (dependendo do modo);
- Nível lógico “0”: -30 a 5 V;
- Nível lógico “1”: 13 a 30V;
- Dados de entrada: 10 Bytes;
- Dados de saída: 10 Bytes;
- Leds de indicação de estado;
- Alimentação: 5 VDC / 80 mA via Bus;
- 24 VDC;
- 40 modos de contagem configuráveis.

Os módulos de contagem utilizam-se para ler os valores dos codificadores.

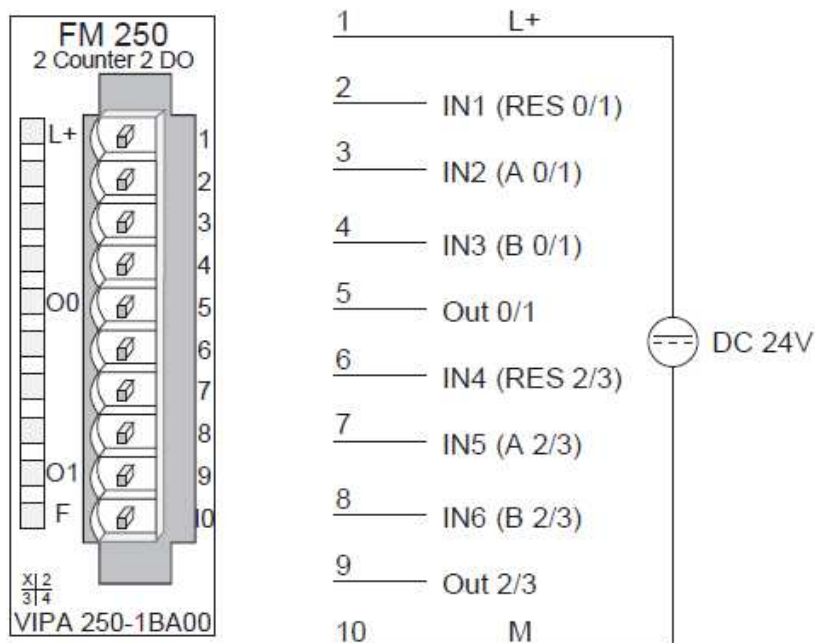


Figura 27: Módulo de contagem (VIPA 250-1BA00)

3.2.9 HMI

Como Interface Homem Máquina utiliza-se um vulgar computador dotado de um ecrã tátil. Para comunicação de dados entre o computador e o PLC utiliza-se um cabo MPI.

3.3 Software

Para que o sistema de controlo funcione de modo a atingir os objetivos propostos é necessário que o controlador execute a aplicação de *software* adequada e que interaja com o interface homem máquina.

3.3.1 Software do autómato

Para elaboração do *software* do controlador consideram-se as entradas e saídas apresentadas na Tabela 4. Estas entradas e saídas fazem o interface entre o controlador, os sensores, os controladores de potência e outros equipamentos.

Tabela 4: Entradas e saídas digitais do PLC

ENTRADAS		SAÍDAS	
Entrada	Descrição	Saída	Descrição
I 0.0	Botão de emergência	Q 0.0	Forward variador eixo X
I 0.1	Botão eixo X+	Q 0.1	Reverse variador eixo X
I 0.2	Botão eixo X-	Q 0.2	Forward variador eixo Y
I 0.3	Botão eixo X rápido	Q 0.3	Reverse variador eixo Y
I 0.4	Botão eixo Y+	Q 0.4	Forward variador eixo Z
I 0.5	Botão eixo Y-	Q 0.5	Reverse variador eixo Z
I 0.6	Botão eixo Y rápido	Q 0.6	Forward variador mesa
I 0.7	Botão eixo Z+	Q 0.7	Reverse variador mesa
I 1.0	Botão eixo Z-	Q 1.0	Frequência 1 variador mesa
I 1.1	Botão eixo Z rápido	Q 1.1	Frequência 2 variador mesa
I 1.2	Botão rodar Mesa +	Q 1.2	Forward variador motor de corte
I 1.3	Botão rodar Mesa -	Q 1.3	Reset variadores
I 1.4	Botão rodar Mesa rápido	Q 1.4	Não utilizada
I 1.5	Botão ligar motor de corte	Q 1.5	Não utilizada
I 1.6	Botão desligar motor de corte	Q 1.6	Não utilizada
I 1.7	Motor de corte em funcionamento	Q 1.7	Não utilizada
I 2.0	Sensor limite eixo X+	Q 2.0	Travão eixo X
I 2.1	Sensor limite eixo X-	Q 2.1	Travão eixo Z
I 2.2	Sensor limite eixo Y+	Q 2.2	Travão mesa
I 2.3	Sensor limite eixo Y-	Q 2.3	Bomba de lubrificação
I 2.4	Sensor limite eixo Z+	Q 2.4	Válvula da água
I 2.5	Sensor limite eixo Z-	Q 2.5	Lâmpada verde
I 2.6	Alarme variador do eixo X	Q 2.6	Lâmpada laranja
I 2.7	Alarme variador do eixo Y	Q 2.7	Lâmpada vermelha
I 3.0	Alarme variador do eixo Z		
I 3.1	Alarme variador da mesa		
I 3.2	Alarme variador de corte		
I 3.3	Alarme disparo disjuntores		
I 3.4	Sensor de fluxo de água		
I 3.5	Pressostato ar		
I 3.6	Nível óleo lubrificação		
I 3.7	Contador energia		

De modo a controlar a máquina desenvolveu-se um programa particionado por várias funções, chamadas de forma sequencial no bloco OB1. A sequência de execução destas funções é apresentada no fluxograma da Figura 28.

Existem três modos principais de funcionamento da máquina:

- **Modo manual**
Neste modo os movimentos dos eixos são efetuados de forma manual através dos botões de comando da botoneira.
- **Modo automático cortar**
Neste modo os eixos Y e Z são movimentados automaticamente, de modo a efetuar cortes no plano YZ. Após cada corte, o eixo Z desloca-se para a posição inicial e é chamada a função FC11 que, em face das medidas das peças inseridas na HMI, vai determinar a próxima posição do eixo X.
- **Modo automático moldar**
Neste modo os eixos Y e Z são movimentados automaticamente de modo a efetuar cortes no plano YZ. Após cada corte, o eixo Z desloca-se para a posição inicial e é chamada a função FC16 que, em face das coordenadas obtidas a partir do desenho do molde, vai determinar a próxima posição dos eixos X e Z.

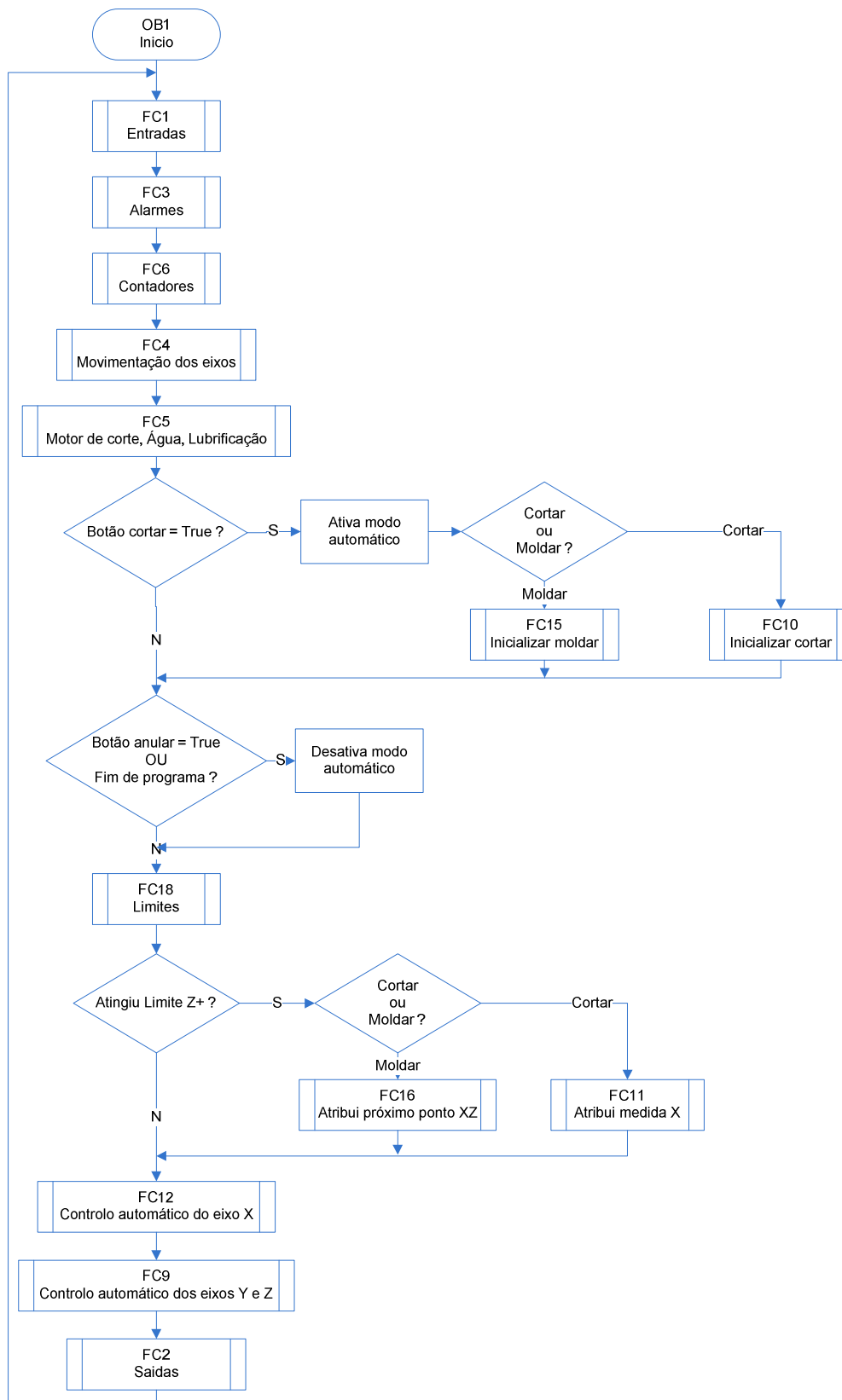


Figura 28: Fluxograma do programa do bloco OB1

Seguidamente apresenta-se a lista descritiva das funções criadas:

FC1 – Entradas

Esta função tem como objetivo tornar mais fácil a adaptação do *software* aos sensores e outros equipamentos disponíveis em máquinas diferentes.

FC2 – Saídas

Esta função tem como objetivo tornar mais fácil a adaptação do *software* aos atuadores e outros equipamentos disponíveis em máquinas diferentes.

FC3 – Alarmes

Nesta função faz-se a monitorização e gestão de alarmes que podem ocorrer, tais como: falta de água, alarme de variadores, disparo de disjuntores térmicos, botões de emergência e outros. Quando se verifica a ocorrência de um alarme todos os movimentos da máquina são interrompidos e os restantes motores são desligados. As mensagens de alarme são mostradas na HMI e o *reset* aos alarmes é efetuado através de um botão na HMI.

FC4 – Movimentação de eixos

Tendo em consideração os botões de comando, os sensores de limite e o modo de funcionamento manual/automático, esta função permite o controlo do movimento de todos os eixos da máquina.

FC5 – Motor de corte / Água / Lubrificação

Esta função controla o motor de corte, a água de arrefecimento do disco de corte e o sistema de lubrificação automática. Em função do diâmetro do disco de corte, é definida a velocidade de rotação do disco. Em funcionamento automático e em intervalos de tempo definidos pelo utilizador é efetuada a lubrificação da máquina.

FC6 – Contadores

Esta função converte o número de impulsos dos contadores dos codificadores dos eixos em décimas de milímetro.

FC7 – Inicializar

Tal como o nome indica, esta função faz a inicialização da máquina de modo a referenciarem-se os eixos X, Y e Z. Esta inicialização é necessária após uma falha de energia, porque poderá existir movimento nalgum dos eixos já com o PLC desligado.

FC8 – Controlo de velocidades

Usando como referência os valores dos potenciômetros ligados às entradas analógicas, são controladas as velocidades de movimentação dos eixos no modo manual e automático. Enquanto não for executada a função inicializar (FC7), as velocidades são reduzidas, uma vez que não se conhecem os limites dos eixos. Em funcionamento, as velocidades também são

reduzidas perto dos extremos físicos da máquina. No modo automático, a velocidade do eixo Y é reduzida antes da inversão de sentido e a do eixo X é reduzida na parte final da medição, de modo a obter-se maior precisão.

Os valores de velocidade pretendidos são enviados para os variadores de velocidade dos motores dos eixos através das saídas analógicas.

FC9 – Corte

Em ciclo automático é efetuado um corte no plano YZ, entre os limites identificados na Figura 29, isto é, desde uma altura inicial (Limite Z+) até à base da pedra (Limite Z-) e entre as posições Limite Y- e Limite Y+ indicadas pelo operador através dos botões de comando.

Este corte é realizado por passagens de alguns milímetros, valor indicado pelo registo descida avanço e descida recuo, sendo o avanço o sentido coincidente com o sentido de rotação do disco de corte.

Pode-se observar o fluxograma simplificado desta função na Figura 30.

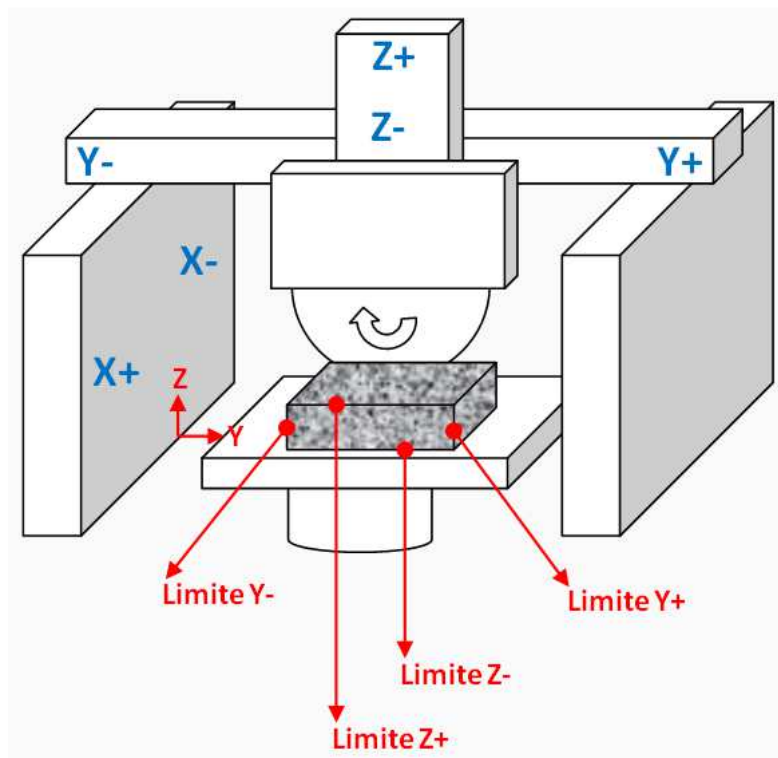


Figura 29: Limites de corte no plano YZ

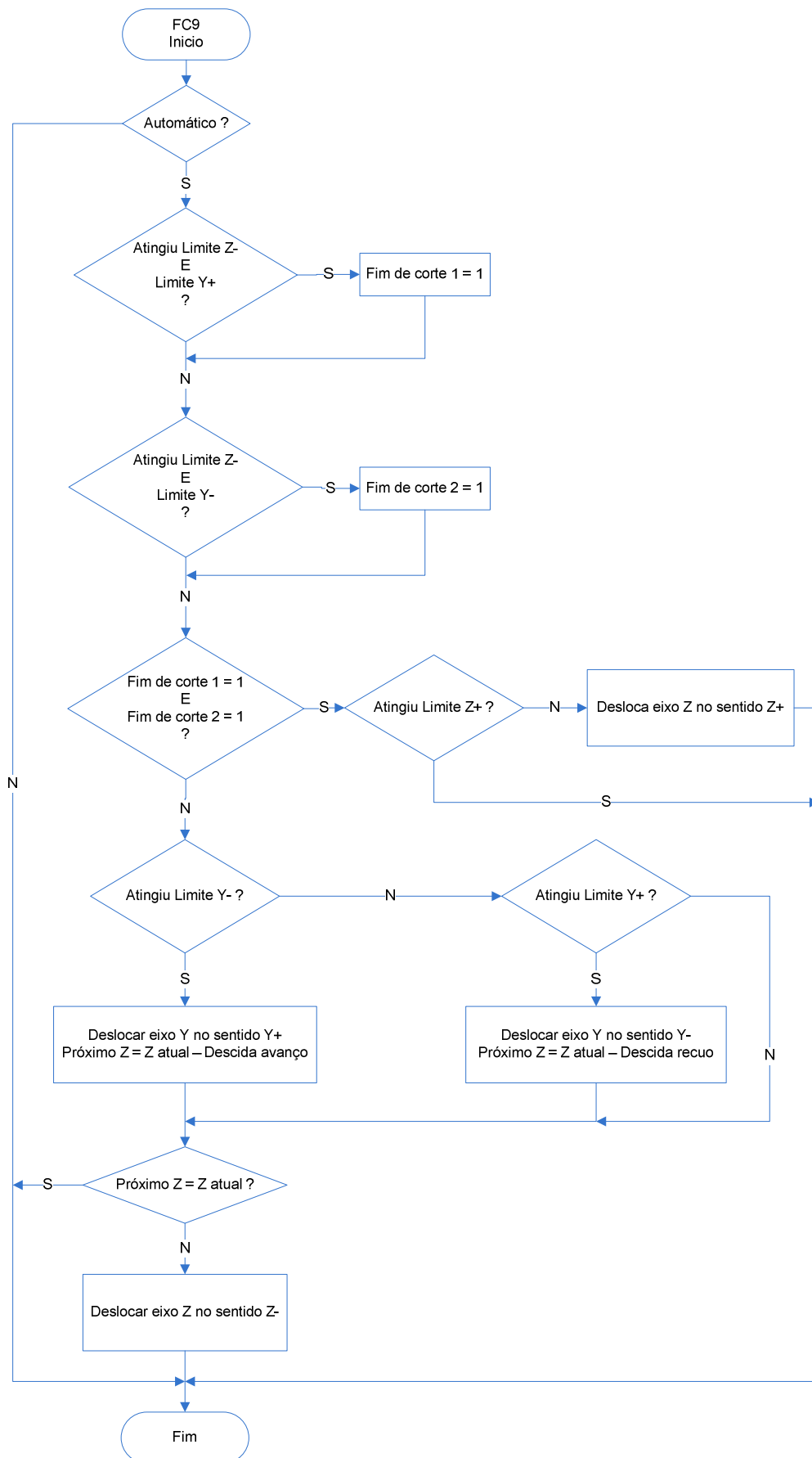


Figura 30: Fluxograma da função Corte FC9

FC10 – Inicializar cortar

Esta função é executada quando se entra no modo automático Cortar, em que o objetivo é atualizar os registos dos cortes que faltam e memorizar as posições iniciais, nomeadamente o Limite Z+ e o Limite Y-.

FC11 – Medidas eixo X

Em face do número de cortes já realizados pela função FC9, FC11 determina a próxima posição do eixo X. A nova posição é determinada pela soma da medida pretendida com a espessura do disco. Se estiver selecionada a medição no sentido X+, esta soma é adicionada à posição atual; se estiver selecionada a medição no sentido X-, esta soma é subtraída à posição atual, obtendo-se a próxima posição do eixo X.

FC12 – Medir eixo X

No modo de ciclo automático e após cada corte, o eixo X sofre um determinado deslocamento calculado por FC11 ou FC16, consoante o modo de funcionamento. De seguida inicia-se um novo corte.

FC13 – Gestão

Esta função realiza diversas tarefas, onde se incluem:

- Atualização dos registos do tempo de funcionamento da máquina para o trabalho em curso, trabalho total e último trabalho;
- Contagem de energia consumida para o trabalho em curso, trabalho total e último trabalho;
- Atualização dos registos do tempo de funcionamento do disco atual e da área por ele cortada. Esta tarefa pode servir para avaliar a qualidade do disco de corte.

FC14 – Posicionar mesa

Em máquinas com mesa automática, esta função permite rodar automaticamente a mesa uma quantidade de graus indicada pelo operador.

FC15 – Inicializar moldar

Esta função é executada quando se entra no modo automático Moldar, tendo como objetivos atualizar o Limite Z-, de acordo com o molde, e memorizar as posições iniciais, nomeadamente o Limite Z+, o Limite Y- e valor de X inicial.

FC16 – Molde XZ

Aqui, em função dos pontos de corte do molde desenhado no computador, são atualizados a nova posição do eixo X e o novo Limite Z- (coordenadas X e Z do molde). Para isso, de cada vez que é chamada esta função, as coordenadas do registo das coordenadas seguintes são transferidas para os registos das coordenadas atuais, e a aplicação HMI que está a ser executada no computador envia as novas coordenadas seguintes.

FC17 – Posicionar eixo X

Esta função permite deslocar automaticamente o eixo X uma distância indicada pelo operador.

FC18 – Limites

Esta função, em execução durante o ciclo automático, deteta os limites “virtuais” dos eixos Y e Z e permite a alteração dos limites de corte do eixo Y.

3.3.2 *Software* HMI

Para o *software* de interface homem-máquina desenvolveu-se uma aplicação em Visual Basic (VB) que é executada num computador com sistema operativo Windows.

O Visual Basic é uma linguagem de programação da Microsoft e é parte integrante do pacote Microsoft Visual Studio. Sendo um aperfeiçoamento do BASIC, a linguagem é dirigida por eventos (*event driven*) e possui também um ambiente de desenvolvimento integrado (IDE — *Integrated Development Environment*) totalmente gráfico, facilitando enormemente a construção da interface das aplicações (GUI — *Graphical User Interface*) [8].

O interface gráfico da máquina em Visual Basic é constituído pelos seguintes ecrãs de aplicação: Ecrã inicial, Corte de Chapas, Cortar por Molde, Desenhar Moldes, Alarmes, Parâmetros, Dados e Configurações.

Seguidamente descrevem-se as funcionalidades mais relevantes associadas aos diversos ecrãs.

Ecrã inicial

Este é o primeiro ecrã do interface gráfico. É constituído por diversos botões de função e botões de interligação a outros ecrãs (Figura 31).

O botão inicializar permite inicializar a máquina de modo a obter as referências dos eixos.

Em termos de aplicação, está aqui implementado o cliente OPC¹³ fazendo-se a ligação ao servidor, atualizando os valores das variáveis das coordenadas dos eixos e monitorizando os alarmes (Ver Código do OPC no Anexo 2).

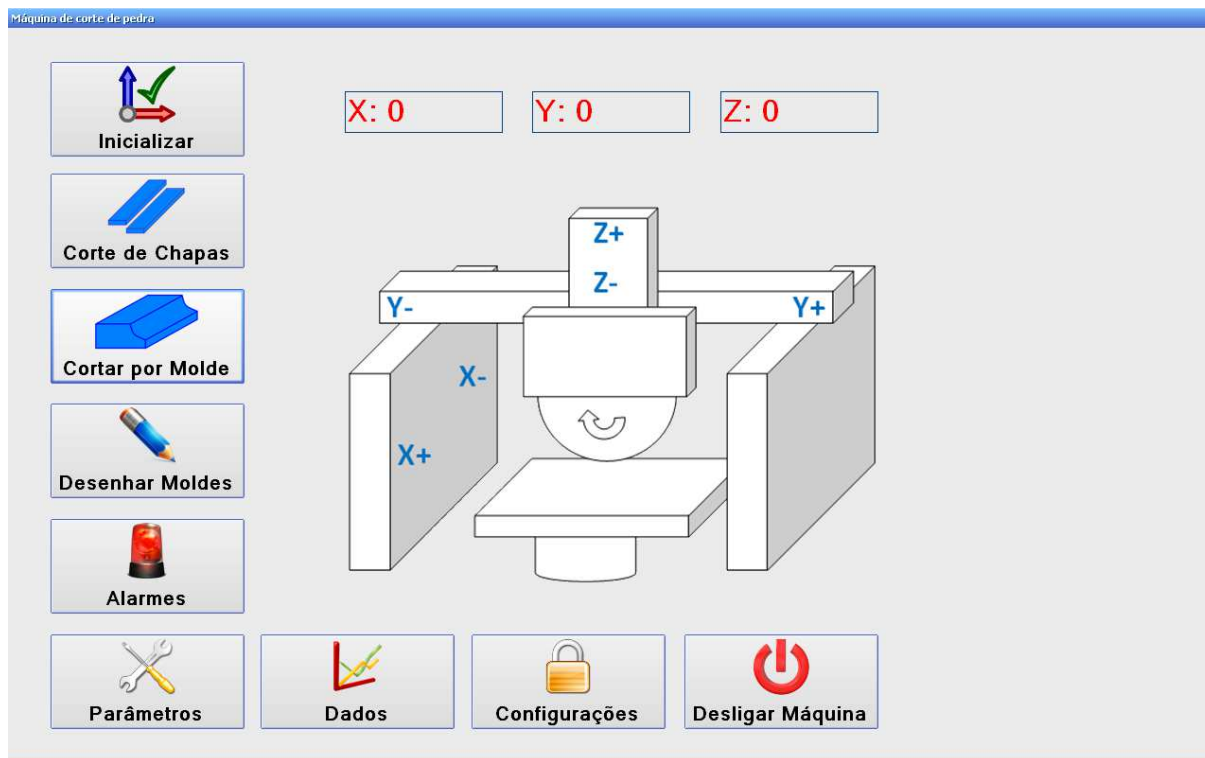


Figura 31: Ecrã inicial

¹³ Object Linking and Embedding (OLE) for Process Control

Desenhar Moldes

Na Figura 32 apresenta-se o interface gráfico simples e intuitivo, através do qual o operador da máquina cria os desenhos dos moldes que pretende maquinar. Os desenhos são retas e curvas, entre coordenadas do plano XZ. Os dados gerados respeitantes aos desenhos são guardados em ficheiros.

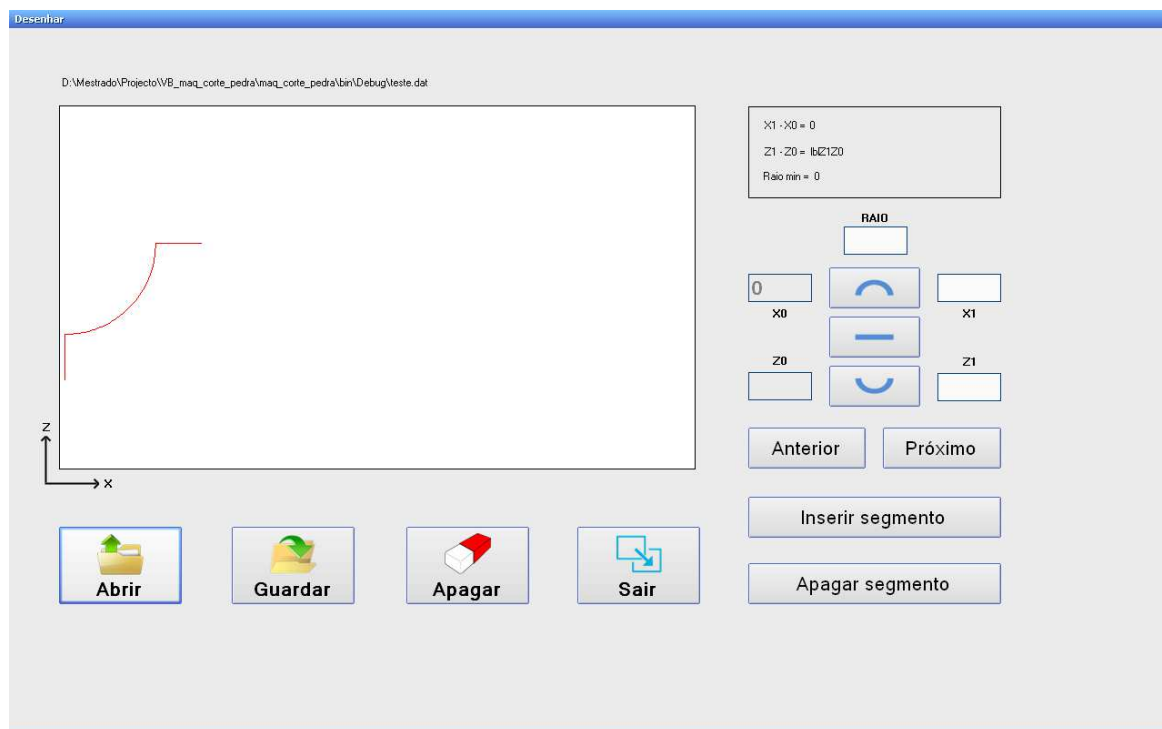


Figura 32: Ecrã Desenhar

Para desenhar um molde insere-se a coordenada do próximo ponto e, no caso de arco de circunferência, insere-se também o raio e depois, através dos botões, selecciona-se o tipo de segmento: reta, arco de circunferência sentido horário ou arco de circunferência sentido anti-horário. Também se podem seleccionar os segmentos já criados através dos botões Anterior e Próximo e pode-se também inserir ou eliminar segmentos através dos respectivos botões.

Para representar o desenho no ecrã foi criada a função desenhar que, usando as funções de cálculo do centro da circunferência e de cálculo da coordenada Z, permite desenhar os segmentos no ecrã. A programação correspondente às funções Código da função desenhar, Código da função determinar centro de circunferência e Código da função de cálculo da coordenada Z pode ser consultada no Anexo 2.

Corte de Chapas

Na Figura 33 apresenta-se o interface para o corte de peças simples. Aqui são introduzidas as medidas e quantidades de peças pretendidas, bem como alguns parâmetros de corte.

Medida	NºPeças	Faltam
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

Total (mm) 0

Apagar

X: 0
Y: 0
Z: 0
M: 0°

Posicionar X
X-
0
X+

Rotação mesa
0
Rodar

Descida avanço (mm)
0

Descida recuo (mm)
0

Altura mínima (mm)
0

Acerto à mesa
Acertar

Sentido da medida
X-

Cortar Anular Sair

Figura 33: Ecrã Corte de Chapa

Os parâmetros “Descida avanço” e “Descida recuo” definem a profundidade de corte em cada passagem do disco. O parâmetro “Altura mínima” define a altura Z até onde vai ser efetuado o corte. O botão “Acertar” permite definir o zero do eixo Z, enquanto que o botão “Sentido da medida”, permite seleccionar em que sentido do eixo X se vai deslocar a máquina para efetuar o corte das peças. O botão “Cortar” serve para dar ordem à máquina de iniciar o ciclo automático. Por sua vez o botão “Anular” serve para anular esse ciclo automático.

Na zona central do ecrã temos visíveis os valores das coordenadas X, Y e Z, bem como a posição da mesa, em graus.

Também é possível deslocar o eixo X de uma determinada distância, através dos botões “Posicionar X-” e “Posicionar X+”. O botão “Rodar” permite rodar a mesa um determinado ângulo.

Cortar por Molde

A função “Cortar por Molde” (Figura 34) permite abrir ficheiros criados pela função “Desenhar Moldes” e determina as coordenadas necessárias à maquinagem da peça. Utilizando-se funções matemáticas, são determinadas as alturas Z, correspondentes a cada ponto em X com um passo de 1mm. Os valores assim obtidos são guardados num *array* de pontos, para posteriormente serem enviados para o autómato.

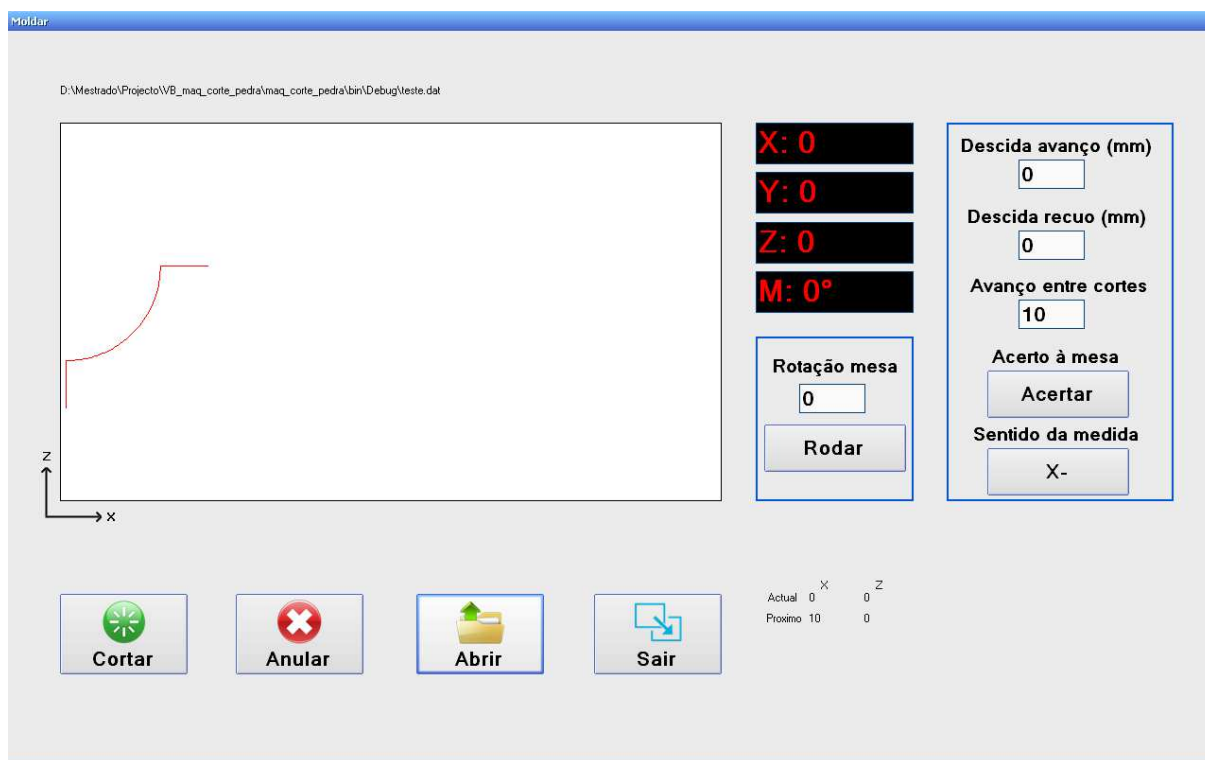


Figura 34: Ecrã Cortar por Molde

Ao abrir-se um ficheiro é criado o objeto “pontos” e de seguida são chamadas as rotinas desenhar e calcular molde, tal como é exemplificado no fluxograma mostrado na Figura 35.

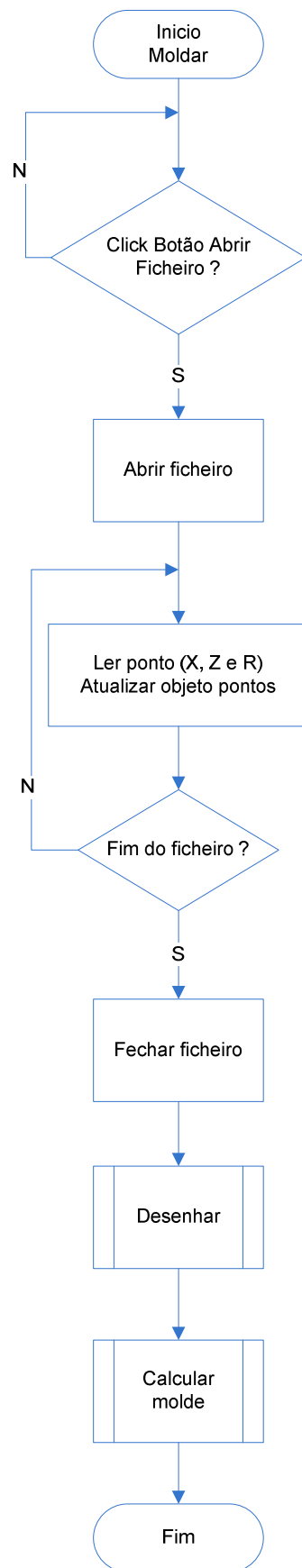


Figura 35: Fluxograma da abertura de ficheiro de molde

A rotina “Desenhar”, tendo como base o objeto “pontos”, verifica, entre cada dois pontos consecutivos, o tipo de segmento (reta ou arco de circunferência) e desenha no ecrã o respetivo segmento (Figura 36).

O código do programa da função desenhar, da função determinar centro de circunferência e da função de cálculo da coordenada Z pode ser consultado no Anexo 2.

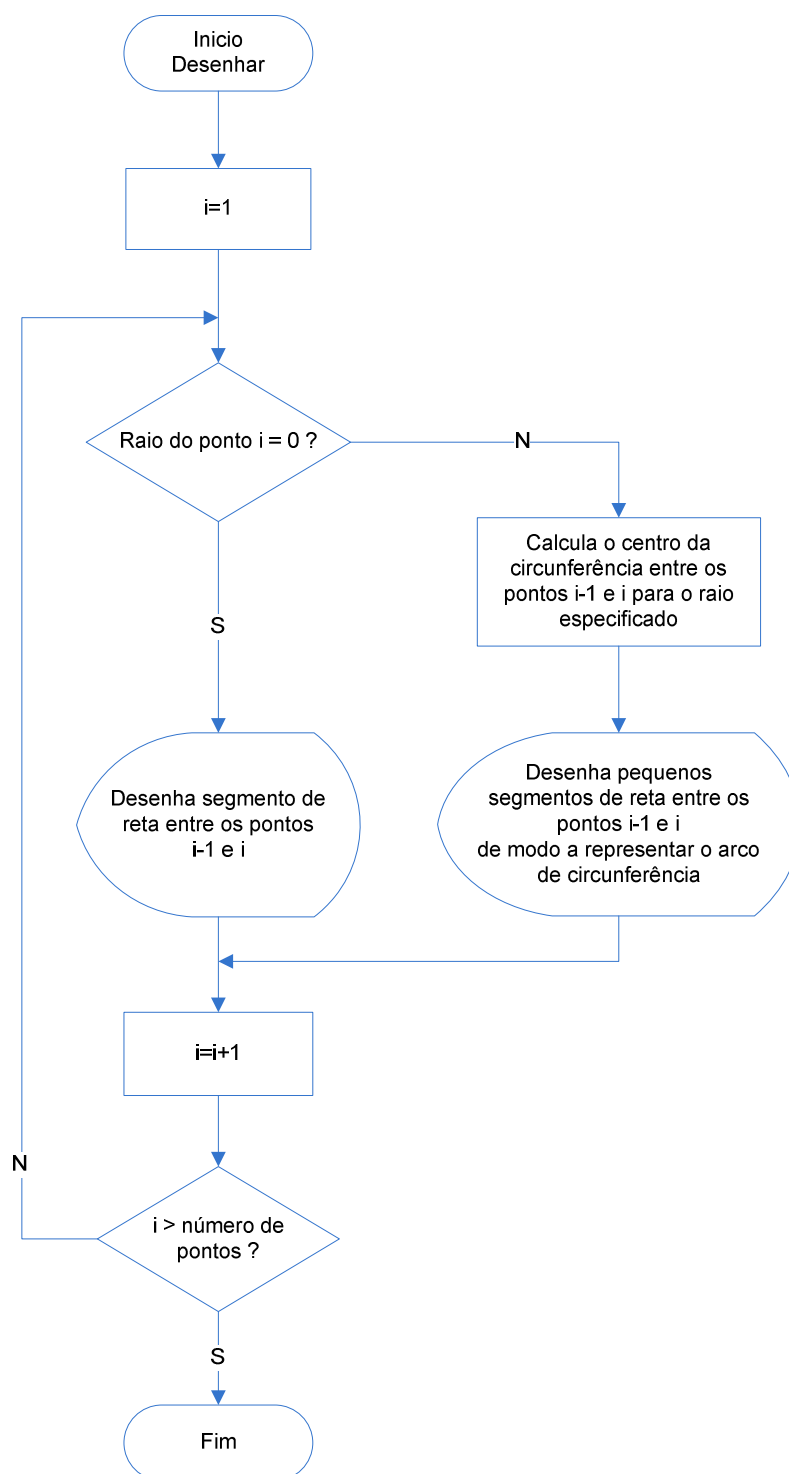


Figura 36: Fluxograma da rotina Desenhar

A rotina “Calcular Molde”, tendo como base o objeto “pontos”, verifica, entre cada dois pontos consecutivos, o tipo de segmento (reta vertical, reta horizontal, reta diagonal ou arco de circunferência) e determina o valor da cota Z para cada valor de X (Figura 37).

A programação da função de cálculo dos pontos de corte do molde pode ser consultada no Anexo 2. As cotas Z são armazenadas num *array* para posteriormente serem enviadas para o autómato.

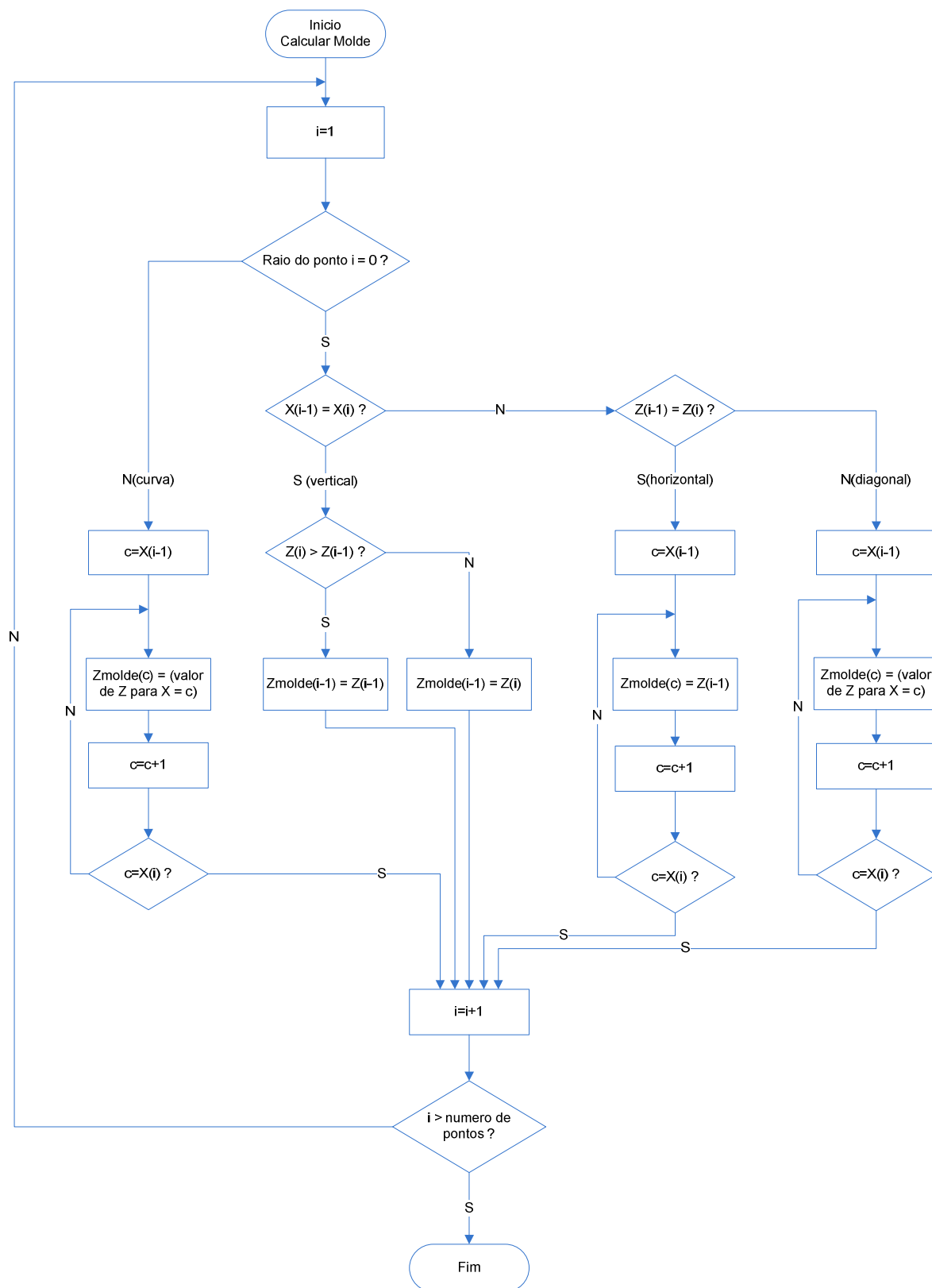


Figura 37: Fluxograma da rotina Calcular Molde

Alarmes

Este ecrã fica visível sempre que existe uma situação de alarme e é mostrada a hora e data, bem como a descrição do alarme.

Parâmetros

Aqui são definidos vários parâmetros da máquina, onde se incluem o diâmetro e a espessura do disco de corte e o tempo entre lubrificações automáticas.

Dados

Neste ecrã são apresentados dados (Figura 38) que permitem controlar os custos de produção, bem como controlar o rendimento da ferramenta diamantada (disco de corte).

São mostrados dados relativos ao trabalho em curso, ao último trabalho e ao total da máquina. As horas do último trabalho permitem, no caso de trabalhos repetitivos, prever o tempo de produção.

Relativamente ao disco de corte é apresentado o seu tempo de funcionamento e a produção aproximada em m².

The screenshot shows the 'Dados' screen with the following elements:

- HORAS DE TRABALHO**
 - Trabalho em curso: 0 min
 - Último trabalho: 0 min
 - Total: 0 h
- DISCO DE CORTE**
 - Horas: 0 h
 - Produção: 0 m²
 - Reset button (red circle with 'RESET' text)
- CONSUMO ENERGÉTICO**
 - Trabalho em curso: 0 Wh
 - Último trabalho: 0 Wh
 - Total: 0 KWh
- Sair** button (blue icon)

Figura 38: Ecrã de Dados

Configurações

Este ecrã de acesso restrito permite definir as relações dos codificadores dos eixos para que, no caso de troca de um codificador por outro com resolução diferente, se possa acertar a relação entre impulsos do codificador e o deslocamento ou rotação do componente.

3.3.3 Interface de comunicação entre o PC e o PLC

Para implementação da HMI é necessário que haja troca de dados entre o PC e o PLC. Normalmente, os PLCs usam protocolos de comunicação proprietários, pelo que surgiu o OPC.

OPC (*OLE for Process Control*) é um padrão industrial publicado para comunicação entre sistemas. As especificações deste padrão são mantidas pela Fundação OPC. Esta Fundação é uma organização dedicada ao desenvolvimento de tecnologias aplicadas à operação entre equipamentos de automação, a fim de criar e gerir especificações que normalizem a comunicação das arquiteturas de acesso a dados *online*, alarmes, registros de eventos, comandos e bases de dados de diferentes equipamentos, de vários fabricantes que comunicam em diferentes protocolos.

O funcionamento do OPC é baseado no OLE (*Object Linking and Embedding*) de componentes orientados a objeto, por meio das tecnologias COM e DCOM da Microsoft, permitindo que as aplicações troquem dados, que podem ser acedidos por um ou mais computadores numa arquitetura cliente/servidor, mesmo que essas aplicações trabalhem sobre sistemas que utilizem protocolos diferentes [9].

Como se pode ver na Figura 39, em termos práticos o OPC consiste num programa servidor que comunica com o PLC ou outros equipamentos, através do protocolo proprietário, e disponibiliza os dados no padrão OPC. O cliente, ao invés de precisar de um *driver* do protocolo proprietário, necessita ter apenas o *driver OPC client* instalado.

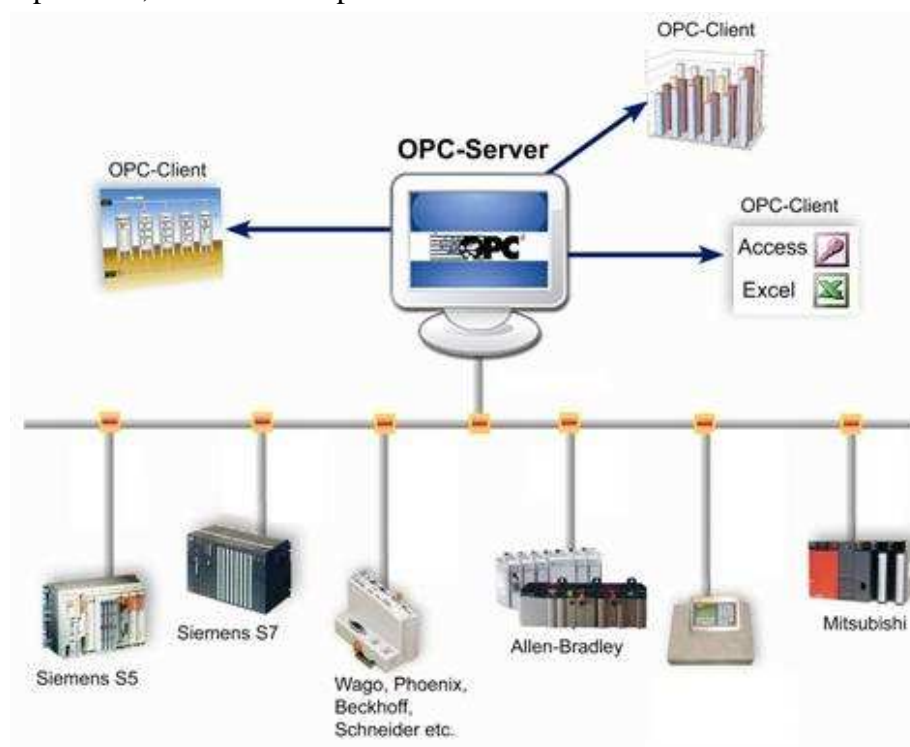


Figura 39: Arquitetura de um sistema OPC

OPC da VIPA

No conjunto de aplicações OPC da VIPA vem incluído o OPC Server, o OPC Editor e *drivers* para o protocolo MPI via COM-Port e protocolo TCP/IP. O OPC Editor é o ambiente de configuração de comunicações, equipamentos e variáveis (*tags*).

3.4 Análise económica

As máquinas de pedra de três eixos possuem estruturas mecânicas robustas e duráveis, mas com a idade os seus sistemas de controlo tornam-se obsoletos.

A estrutura mecânica representa grande parte do valor destas máquinas, pelo que surgiu a necessidade deste projeto de modernização do sistema de automação, mantendo toda a estrutura mecânica e respetivos acionamentos mecânicos.

Para a modernização do sistema de automação são necessários novos equipamentos. Os custos dos equipamentos mais relevantes apresentam-se na Tabela 5:

Tabela 5: Custo dos equipamentos

Descrição	Referência	Quantidade	Valor total
PLC VIPA	CPU 214 (214-1BA02)	1	344 €
	DI 16 (221-1BH30)	2	180 €
	DO 16 (222-1BH30)	1	103 €
	DO 8 (222-1HF00)	1	76 €
	AI 4x16 bit (231-1BD53)	1	191 €
	AO 4x12 bit (232-1BD51)	1	191 €
	2 counter (250-1BA00)	2	478 €
Cabo de comunicação RS232-MPI/PPI	950-0KB10	1	242 €
Variador de frequência da cabeça de corte	KEB F5.C 22kW	1	1100 €
Variadores de frequência dos eixos	KEB F5.C 1,5kW	3	1020 €
Sensores indutivos	Omron E2A-M18KS08-WP-B1 2M	6	192 €
Codificadores 1000imp		3	480 €
Sensor de fluxo de água	IFM SI5010	1	360 €
Pressostato		1	56 €
Computador com ecrã tátil		1	780 €
Quadro	1400x800x400 mm	1	627 €
Transformador	400V/24V/160VA	1	57 €
Transformador	400V/230V/400VA	1	110 €
Fonte de alimentação 24Vdc	230V/24Vdc/5A	1	72 €
Interruptor de corte geral	80A	1	52 €
Disjuntores	2 a 6A Bipolares	6	162 €
Disjuntor motor 60A		1	172 €
Disjuntor motor 6A		3	156 €
Contactador geral		1	246 €
Outros equipamentos			500 €
Cablagens			230 €
TOTAL:			8.177 €

Somando ao custo dos equipamentos o custo do *software* (800€) e os custos de instalação e colocação do novo sistema em funcionamento (1000€), pode-se dizer que a modernização proposta tem um custo aproximado de 10.000€, o que representa cerca de 14% do custo de uma máquina nova com funcionalidades idênticas.

4. Resultados

Para verificar o correto funcionamento dos programas desenvolvidos, o PLC foi montado numa máquina com os requisitos mínimos para efetuar alguns testes. A máquina utilizada está apresentada na Figura 40. Para esta máquina as dimensões máximas das peças a cortar são de 3000 x 1800 x 300 mm.

Após efetuadas todas as ligações, verificou-se o funcionamento dos sensores fins de curso e testaram-se manualmente os movimentos de todos os eixos. Também se verificou o correto funcionamento da cabeça de corte.



Figura 40: Máquina de corte de pedra

Quanto aos modos automáticos de funcionamento temos o modo de “Corte de Chapas” e o modo “Cortar por Molde”.

No modo “Corte de Chapas” introduziram-se as medidas das peças a cortar, tal como ilustrado na Figura 41. Após ligar o motor da cabeça de corte e premindo o botão “Cortar” a máquina efetuou o corte das peças, como se mostra na Figura 42.

Neste modo de funcionamento verificou-se uma redução significativa do tempo de corte comparativamente com o antigo sistema de controlo, principalmente devido ao fato de nos limites de corte os eixos Y e Z se movimentarem em simultâneo.

Para quantificar aproximadamente essa redução de tempo, utilizando velocidades e profundidades de corte dentro dos valores médios, efetuou-se um ensaio numa peça com altura $Z = 300$ mm e comprimento $Y = 2000$ mm. Com passagens de 5 mm (Descida Corte e Descida Recuo) o eixo Z demora 2,3 segundos a posicionar-se, então considerando a altura de corte de 300 mm temos $300/5=60$ passagens o que implica um tempo total de $60 \times 2,3=138$ segundos. Numa peça com 2000 mm de comprimento temos um tempo total de corte de 720 segundos, concluindo-se que para este exemplo concreto, deslocando os eixos Y e Z em simultâneo obtém-se uma redução de 16% no tempo de corte.



Figura 41: Ecrã Corte de Chapa



Figura 42: Máquina a efetuar corte de chapa

No modo “Cortar por Molde” usando o ecrã mostrado na Figura 43, abriu-se um desenho previamente elaborado. Após ligar o motor da cabeça de corte e premindo o botão Cortar a máquina iniciou a maquinação da peça, como se pode ver na Figura 44. Após terminada a tarefa, a peça ficou com o aspeto que se pode ver na Figura 45. Posteriormente a peça segue para outra secção da fábrica para acabamento.

Usando este processo de fabrico pode-se em alguns segundos alterar o perfil de corte da peça a maquinar. Outros sistemas presentes em máquinas com alguns anos implicam a elaboração de um molde de madeira ou a determinação manual das coordenadas de corte, tarefas que demoram vários minutos. Comparando estes processos de fabrico verifica-se uma drástica diminuição do tempo de projeto da peça a maquinar, tornando esta máquina bastante flexível e fácil de operar.

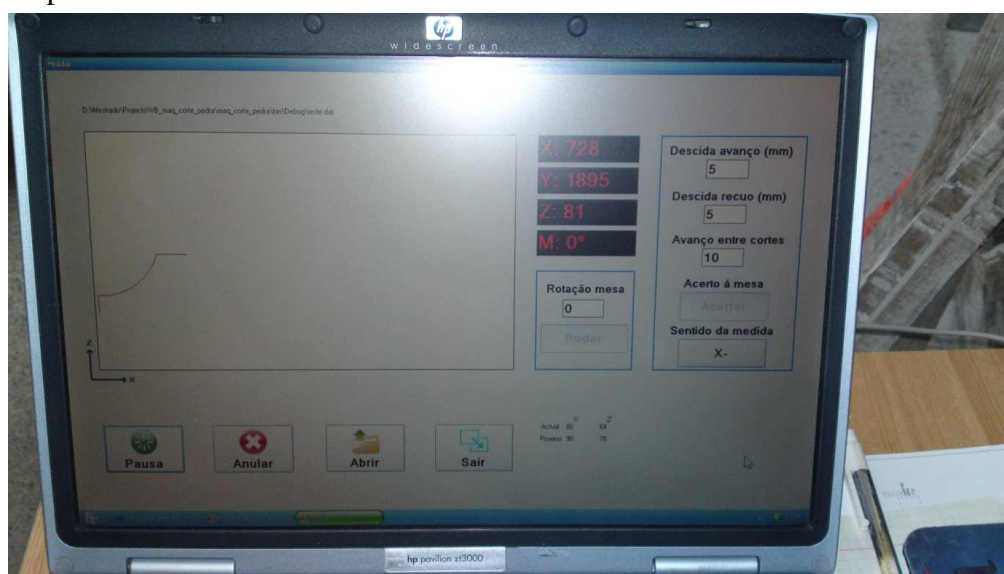


Figura 43: Ecrã Cortar por Molde



Figura 44: Máquina a efetuar a maquinação da peça pretendida



Figura 45: Peça maquinada

5. Conclusões e desenvolvimentos futuros

O interesse pela execução deste projeto surgiu porque ao longo de alguns anos em que tenho dado assistência técnica a máquinas de corte de pedra identifiquei a necessidade de poder oferecer uma solução para remodelação de máquinas que permita o corte de pedra de acordo com um perfil desenhado.

Com o objetivo de modernizar uma máquina existente fez-se a sua automatização utilizando variadores de velocidade, codificadores, diversos sensores, um PLC e uma HMI. Foram elaborados *softwares* para o PLC e HMI que permitem implementar as funções mais comuns de corte de pedra. Adotou-se esta abordagem em vez do CNC, pois este é demasiado dispendioso para as tarefas pretendidas e com recursos que não seriam utilizados devido a limitações mecânicas do tipo de máquinas em causa.

Utilizando como ferramenta de programação para o interface de utilizador o Visual Basic, conseguiu-se uma interface simples, apelativa e de fácil utilização, tornando a máquina fácil de operar.

Após os ensaios efetuados verificou-se que este trabalho implementa as funcionalidades fundamentais para o correto funcionamento da máquina automatizada.

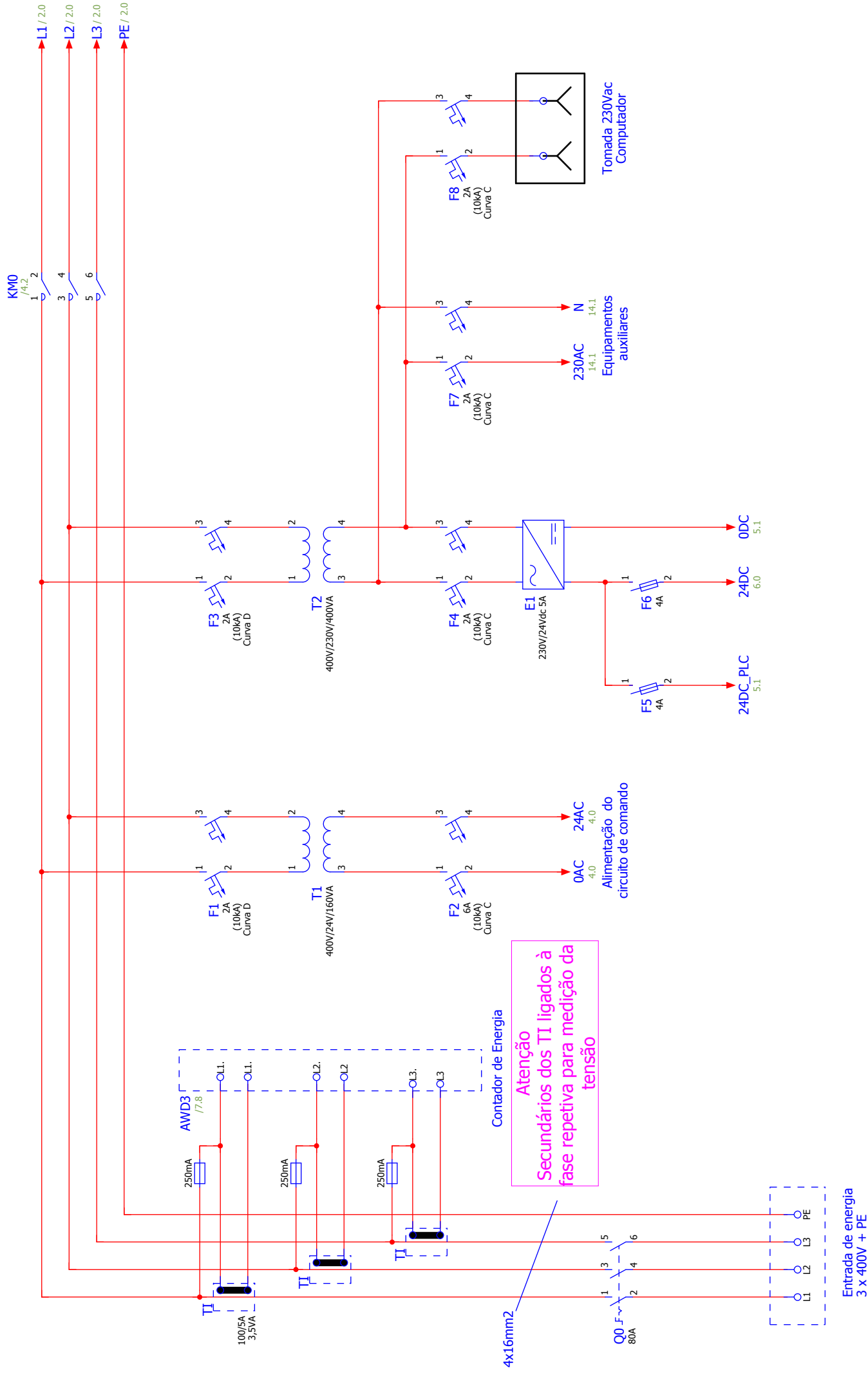
No entanto, face à utilização prática do equipamento desenvolvido é natural que se venha a verificar a necessidade de melhorias em alguns aspetos. Já foram identificadas algumas melhorias a fazer, principalmente o aperfeiçoamento do algoritmo de determinação dos pontos de corte do molde e a validação dos dados introduzidos pelo operador. Deste modo, pode-se assumir que existirão ainda trabalhos futuros a serem desenvolvidos, para dar continuidade ao que aqui foi iniciado.

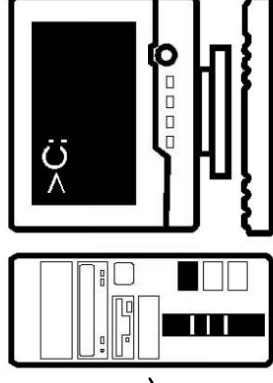
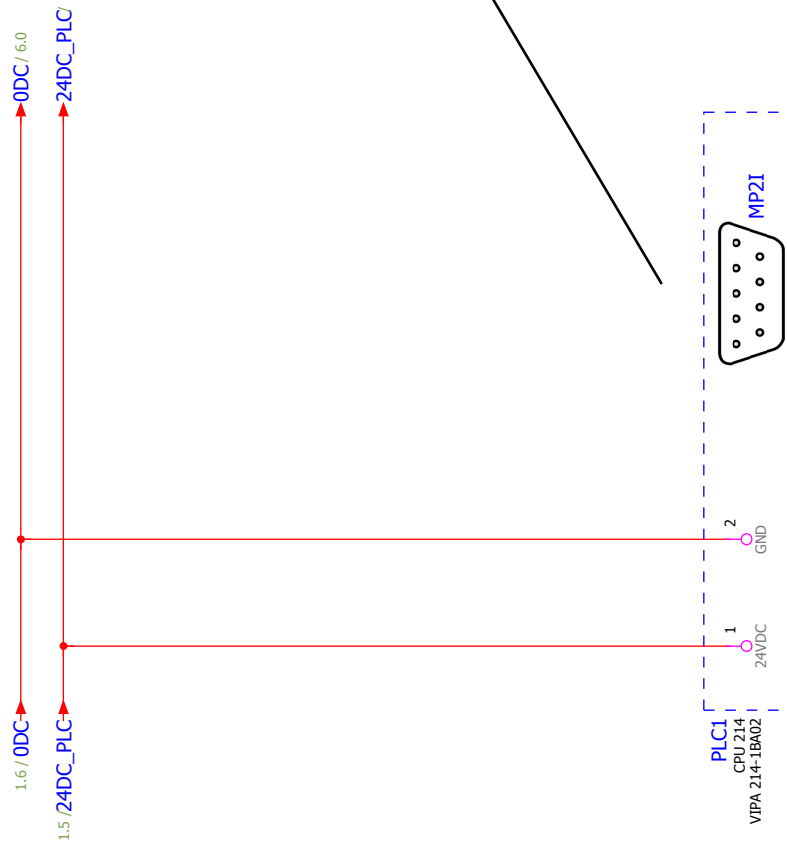
REFERÊNCIAS

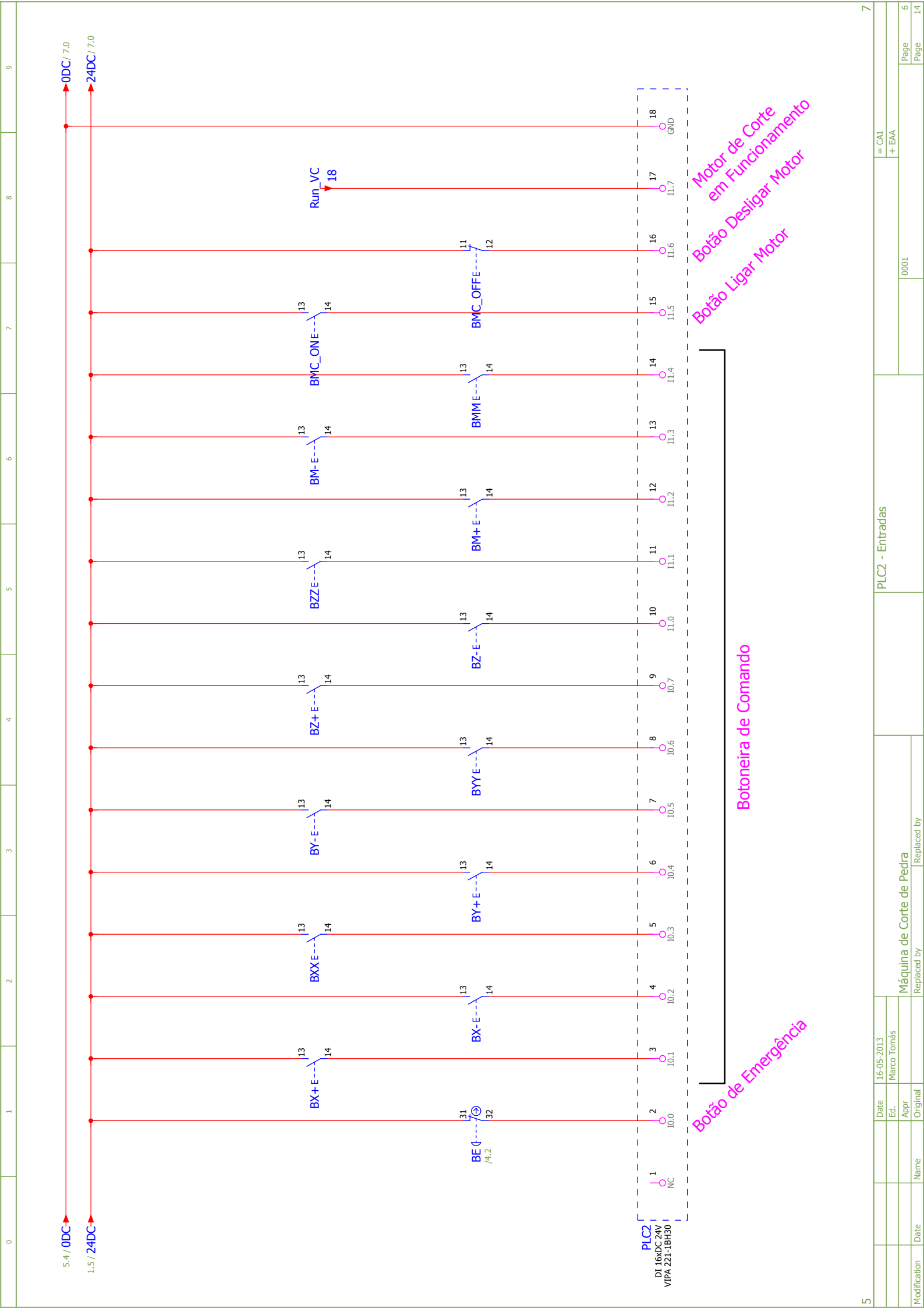
- [1] Filipe Palma, “Pedra portuguesa quer ‘conquistar’ mercados internacionais,” Entrevista Rádio Renascença, 16-Jul-2012.
- [2] Hermann Kopetz, *Real-Time Systems: Design Principles for Distributed Embedded Applications*, 1^a ed. Kluwer Academic Publishers, 1997.
- [3] David Bailey and Edwin Wright, *Practical SCADA for Industry*, 1st ed. Elsevier Science, 2003.
- [4] Hugh Jack, *Automating Manufacturing Systems with PLCs*, 4.7 ed. Lulu.com, 2005.
- [5] Siemens Engenharia e Service, “Arquitetura dos Blocos.” SITRAIN Training for Automation and Drives, 26-Aug-2011.
- [6] Carlos Relvas, *Controlo Numérico Computorizado Conceitos Fundamentais*, Publindústria, 2000.
- [7] *Mecatrônica Atual*, 3^a ed. Editora Saber Ltda, 2002.
- [8] Thearon Willis and Bryan Newsome, *Visual Basic 2010*, 1^a ed. Wrox, 2010.
- [9] Frank Iwanitz and Jorgen Lange, *OPC - Fundamentals Implementation and Application*. Laxmi Publications, 2010.

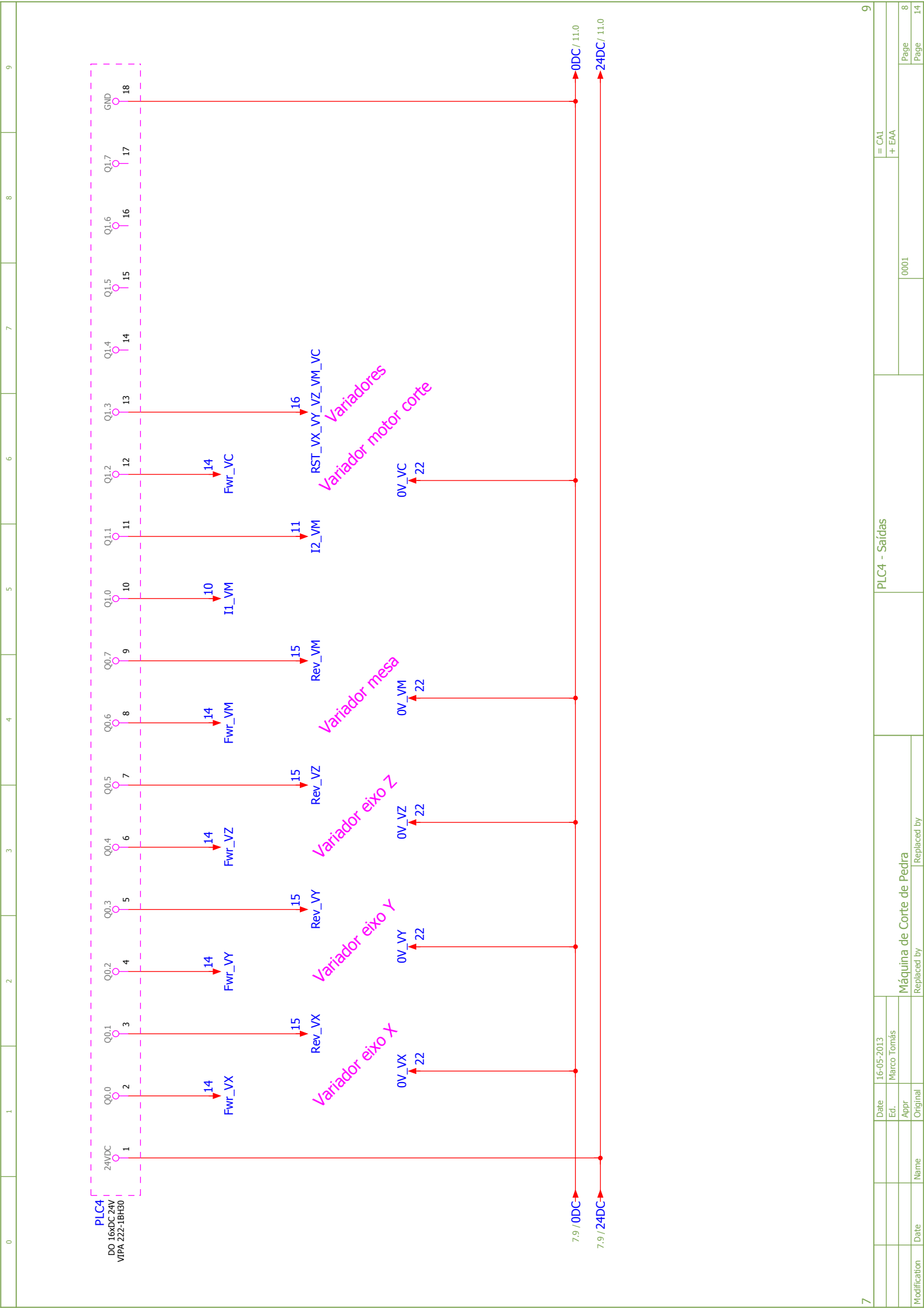
ANEXO 1

Esquema elétrico de potência e comando da máquina









7

9

Date

16-05-2013

Ed.

Marco Tomás

Appr

Name

Date

Original

Replaced by

Máquina de Corte de Pedra

Replaced by

PLC4 - Saídas

0001

Page

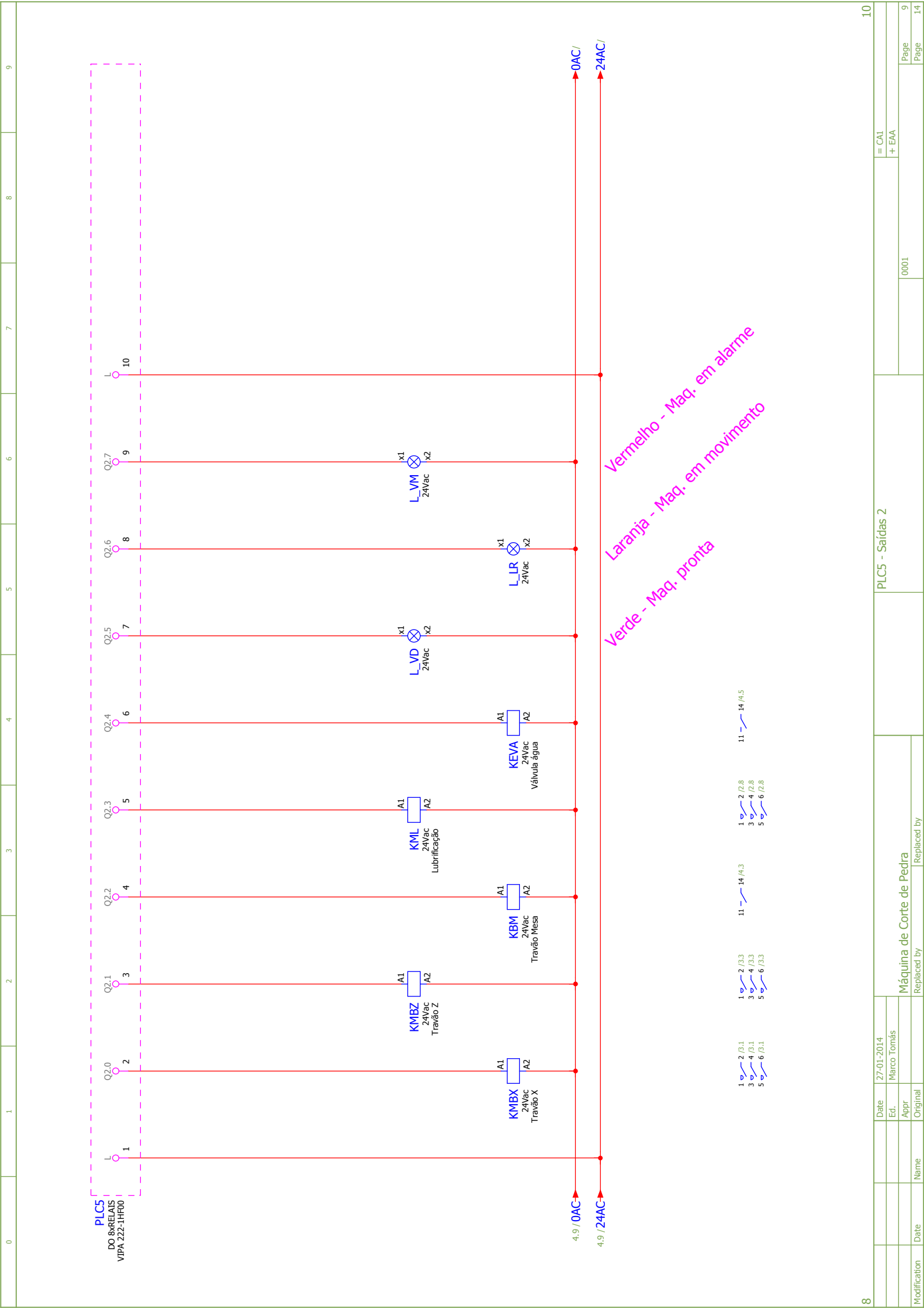
8

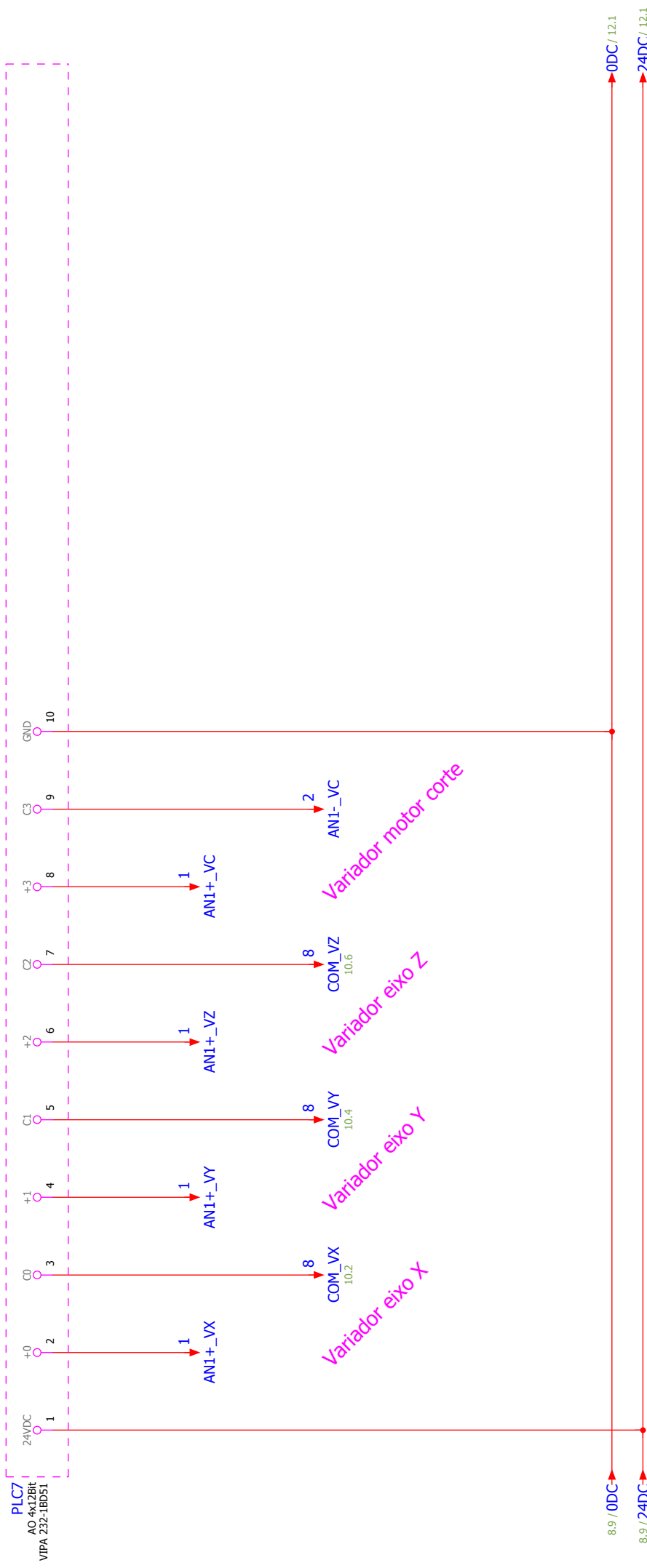
Page

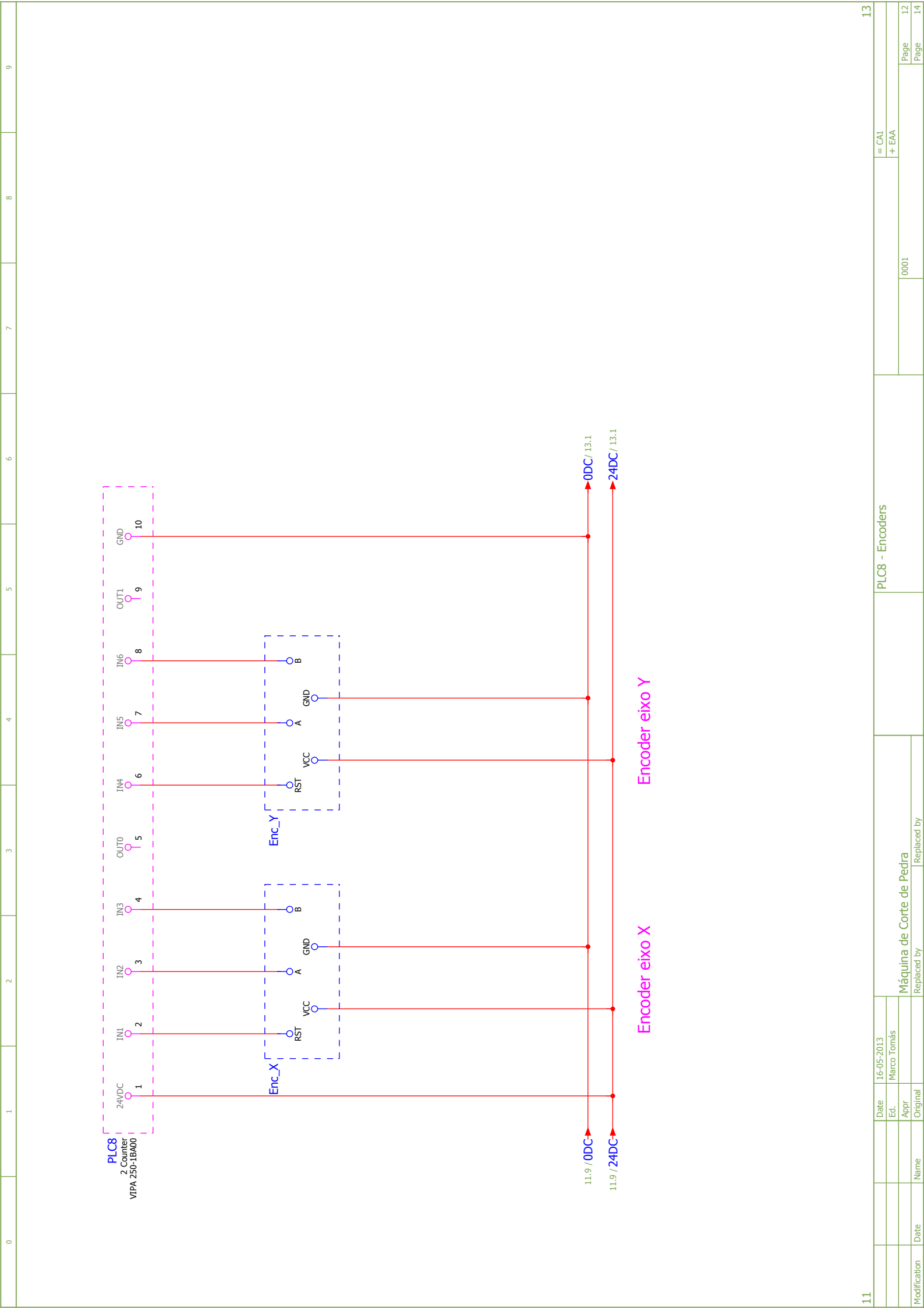
14

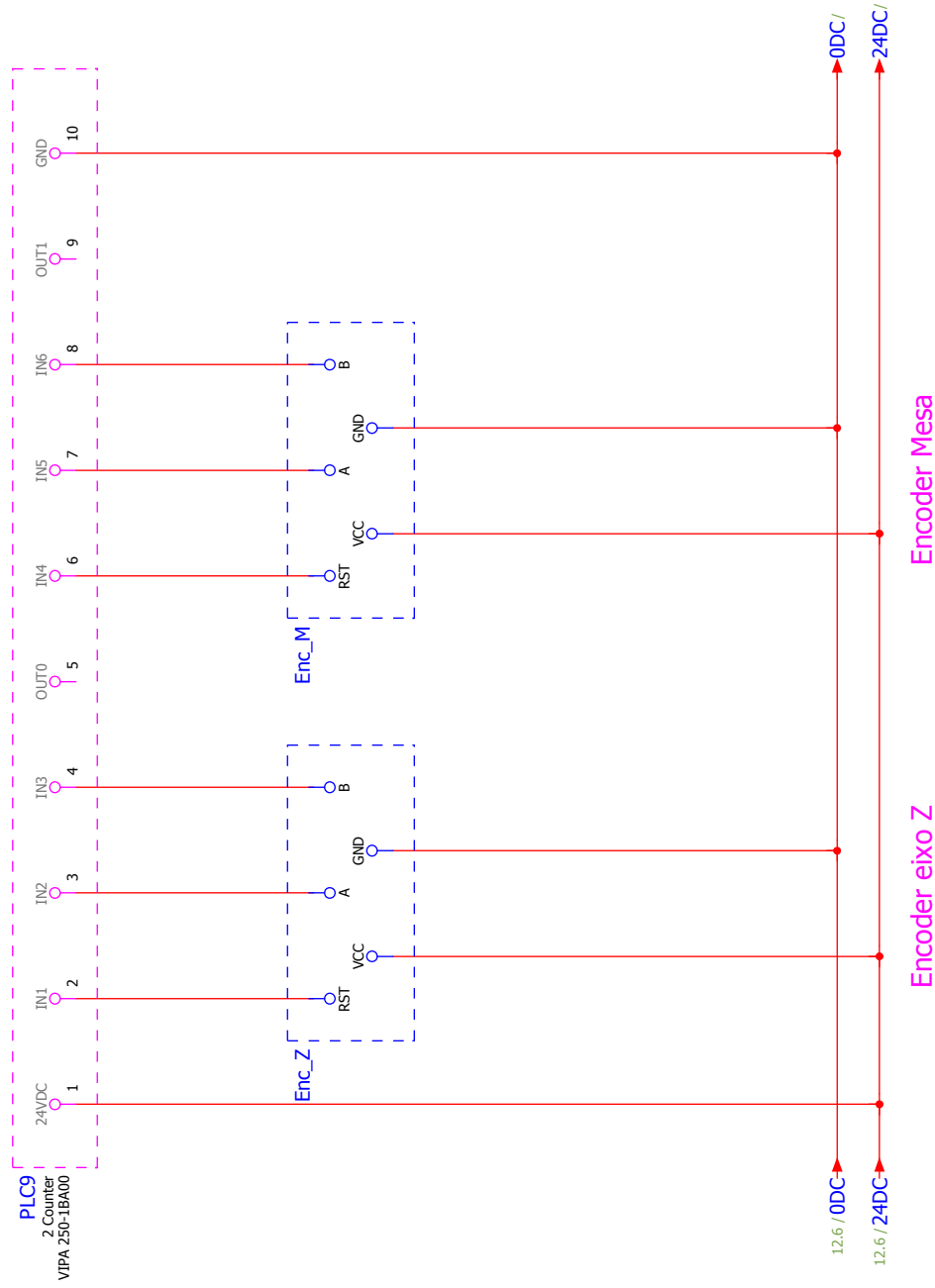
9

9









ANEXO 2

Código do OPC

```
'*****
'*                                     OPC                                     *
'*****

Inherits System.Windows.Forms.Form

Dim WithEvents OPCMyServer As OPCAutomation.OPCServer
Dim WithEvents OPCMyGroups As OPCAutomation.OPCGroups

'Numero de variaveis de cada grupo
Const ITEMMAX_ger As Integer = 2
Const ITEMMAX_alr As Integer = 12
Const ITEMMAX_val As Integer = 8

Const STR_OPC_INI_ger As String = "OPC_geral.ini"
Const STR_OPC_INI_alr As String = "OPC_alarmes.ini"
Const STR_OPC_INI_val As String = "OPC_valores.ini"

Dim WithEvents OPCMyGroup_ger As OPCAutomation.OPCGroup
Dim WithEvents OPCMyGroup_alr As OPCAutomation.OPCGroup
Dim WithEvents OPCMyGroup_val As OPCAutomation.OPCGroup

Dim OPCMyItems_ger As OPCAutomation.OPCItems
Dim OPCMyItems_alr As OPCAutomation.OPCItems
Dim OPCMyItems_val As OPCAutomation.OPCItems

Dim OPCMyItem_ger As OPCAutomation.OPCItem
Dim OPCMyItem_alr As OPCAutomation.OPCItem
Dim OPCMyItem_val As OPCAutomation.OPCItem

Dim sItemName_ger(ITEMMAX_ger) As String
Dim sItemName_alr(ITEMMAX_alr) As String
Dim sItemName_val(ITEMMAX_val) As String

Dim cH_ger(ITEMMAX_ger) As Integer
Dim cH_alr(ITEMMAX_alr) As Integer
Dim cH_val(ITEMMAX_val) As Integer

Dim sH_ger(ITEMMAX_ger) As Integer
Dim sH_alr(ITEMMAX_alr) As Integer
Dim sH_val(ITEMMAX_val) As Integer

'Array com os valores de cada grupo
Public oGeral(ITEMMAX_ger) As Object
Public oAlarmes(ITEMMAX_alr) As Object
Public oValores(ITEMMAX_val) As Object

Dim txtItemName_ger(ITEMMAX_ger - 1) As System.Windows.Forms.Label
Dim txtItemName_alr(ITEMMAX_alr - 1) As System.Windows.Forms.Label
Dim txtItemName_val(ITEMMAX_val - 1) As System.Windows.Forms.Label

'Nome do servidor OPC
Public nome_servidor As String = "VIPA.OPCServer.1"
```

```

Private Sub primeira_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Dim i%

    '***** Cria variaveis *****
    For i = 0 To ITEMMAX_ger - 1
        txtItemName_ger(i) = New System.Windows.Forms.Label
        Me.Controls.Add(txtItemName_ger(i))
    Next

    For i = 0 To ITEMMAX_alr - 1
        txtItemName_alr(i) = New System.Windows.Forms.Label
        Me.Controls.Add(txtItemName_alr(i))
    Next

    For i = 0 To ITEMMAX_val - 1
        txtItemName_val(i) = New System.Windows.Forms.Label
        Me.Controls.Add(txtItemName_val(i))
    Next

    '***** Ler o ficheiro das tags *****
    FileOpen(1, STR_OPC_INI_ger, OpenMode.Input)
    i = 0
    Do While Not EOF(1) ' loop until end of file
        If i > ITEMMAX_ger Then
            Exit Do
        End If
        Input(1, txtItemName_ger(i).Text)
        i = i + 1
    Loop
    FileClose(1)

    FileOpen(1, STR_OPC_INI_alr, OpenMode.Input)
    i = 0
    Do While Not EOF(1) ' loop until end of file
        If i > ITEMMAX_alr Then
            Exit Do
        End If
        Input(1, txtItemName_alr(i).Text)
        i = i + 1
    Loop
    FileClose(1)

    FileOpen(1, STR_OPC_INI_val, OpenMode.Input)
    i = 0
    Do While Not EOF(1) ' loop until end of file
        If i > ITEMMAX_val Then
            Exit Do
        End If
        Input(1, txtItemName_val(i).Text)
        i = i + 1
    Loop
    FileClose(1)

    '***** Efectuar ligacao ao OPCServer *****
    Dim ServerHandles As Array

```

```

Dim Errors As Array

OPCMyServer = New OPCAutomation.OPCServer
OPCMyServer.Connect(nome_servidor, "")
OPCMyGroups = OPCMyServer.OPCGroups

'Grupo geral
OPCMyGroup_ger = OPCMyGroups.Add("Group1")
OPCMyGroup_ger.UpdateRate = 1000      ' UpdateRate is 1000msec
OPCMyGroup_ger.IsActive = False
OPCMyGroup_ger.IsSubscribed = OPCMyGroup_ger.IsActive
OPCMyItems_ger = OPCMyGroup_ger.OPCItems

For i = 1 To ITEMMAX_ger
    sItemName_ger(i) = txtItemName_ger(i - 1).Text
    cH_ger(i) = i
Next
OPCMyItems_ger.AddItems(ITEMMAX_ger, sItemName_ger, cH_ger, ServerHandles,
Errors)
For i = 1 To ITEMMAX_ger
    If Errors(i) = 0 Then
        sH_ger(i) = ServerHandles(i)
    Else
        MsgBox("Registering Items GERAL is failed : " + sItemName_ger(i))
    End If
Next

'Grupo alarmes
OPCMyGroup_alr = OPCMyGroups.Add("Group2")
OPCMyGroup_alr.UpdateRate = 100      ' UpdateRate is 100msec
OPCMyGroup_alr.IsActive = False
OPCMyGroup_alr.IsSubscribed = OPCMyGroup_alr.IsActive
OPCMyItems_alr = OPCMyGroup_alr.OPCItems

For i = 1 To ITEMMAX_alr
    sItemName_alr(i) = txtItemName_alr(i - 1).Text
    cH_alr(i) = i
Next
OPCMyItems_alr.AddItems(ITEMMAX_alr, sItemName_alr, cH_alr, ServerHandles,
Errors)
For i = 1 To ITEMMAX_alr
    If Errors(i) = 0 Then
        sH_alr(i) = ServerHandles(i)
    Else
        MsgBox("Registering Items ALR is failed : " + sItemName_alr(i))
    End If
Next

'Grupo valores
OPCMyGroup_val = OPCMyGroups.Add("Group3")
OPCMyGroup_val.UpdateRate = 100      ' UpdateRate is 100msec
OPCMyGroup_val.IsActive = False
OPCMyGroup_val.IsSubscribed = OPCMyGroup_val.IsActive
OPCMyItems_val = OPCMyGroup_val.OPCItems

For i = 1 To ITEMMAX_val
    sItemName_val(i) = txtItemName_val(i - 1).Text
    cH_val(i) = i

```

```

Next
OPCMyItems_val.AddItem(ITEMMAX_val, sItemName_val, cH_val, ServerHandles,
Errors)
For i = 1 To ITEMMAX_val
    If Errors(i) = 0 Then
        sH_val(i) = ServerHandles(i)
    Else
        MsgBox("Registering Items VAL is failed : " + sItemName_val(i))
    End If
Next

'***** Activa leitura ciclica de valores *****
OPCMyGroup_alr.IsActive = True
OPCMyGroup_alr.IsSubscribed = OPCMyGroup_alr.IsActive

OPCMyGroup_val.IsActive = True
OPCMyGroup_val.IsSubscribed = OPCMyGroup_val.IsActive

End Sub

Protected Overrides Sub Finalize()
    MyBase.Finalize()
End Sub

'***** Actualizar valores no PC *****

Private Sub OPCMyGroup_alr_DataChange(ByVal TransactionID As Integer, ByVal
NumItems As Integer, ByRef ClientHandles As System.Array, ByRef ItemValues As
System.Array, ByRef Qualities As System.Array, ByRef TimeStamps As System.Array)
Handles OPCMyGroup_alr.DataChange
    Dim i%
    For i = 1 To NumItems
        oAlarmes(ClientHandles(i)) = ItemValues(i)
        If oAlarmes(ClientHandles(i)) = True Then
            alarmes.Show()
            alarmes.BringToFront()
        End If
    Next i
    alarmes.update_ListBoxAlr()
End Sub

Private Sub OPCMyGroup_val_DataChange(ByVal TransactionID As Integer, ByVal
NumItems As Integer, ByRef ClientHandles As System.Array, ByRef ItemValues As
System.Array, ByRef Qualities As System.Array, ByRef TimeStamps As System.Array)
Handles OPCMyGroup_val.DataChange
    Dim i%
    For i = 1 To NumItems
        oValores(ClientHandles(i)) = ItemValues(i)
    Next i

    actualiza_coordenadas()

    If oValores(3) = True Then
        btnInicializar.Text = "Maq. Inicializada"
    Else
        btnInicializar.Text = "Inicializar"
    End If
End Sub

```

```

End If

If oValores(1) = True Then
    cortar.btnCortar.Text = "Pausa"
    moldar.btnCortar.Text = "Pausa"

    cortar.btnAcertar.Enabled = False
    cortar.btnRodarMesa.Enabled = False
    cortar.btnPosXp.Enabled = False
    cortar.btnPosXn.Enabled = False
    moldar.btnAcertar.Enabled = False
    moldar.btnRodarMesa.Enabled = False
Else
    cortar.btnCortar.Text = "Cortar"
    moldar.btnCortar.Text = "Cortar"

    cortar.btnAcertar.Enabled = True
    cortar.btnRodarMesa.Enabled = True
    cortar.btnPosXp.Enabled = True
    cortar.btnPosXn.Enabled = True
    moldar.btnAcertar.Enabled = True
    moldar.btnRodarMesa.Enabled = True
End If

End Sub

Public Sub atualiza_coordenadas()
    Dim X, Y, Z, M As String
    X = Math.Round(UnsignedToLong(oValores(4)) / 10)
    Y = Math.Round(UnsignedToLong(oValores(5)) / 10)
    Z = Math.Round(UnsignedToLong(oValores(6)) / 10)
    M = Math.Round(UnsignedToLong(oValores(7)) / 10)

    txtPosX.Text = "X: " + X
    txtPosY.Text = "Y: " + Y
    txtPosZ.Text = "Z: " + Z

    cortar.txtPosX.Text = "X: " + X
    cortar.txtPosY.Text = "Y: " + Y
    cortar.txtPosZ.Text = "Z: " + Z
    cortar.txtPosM.Text = "M: " + M + "°"

    moldar.txtPosX.Text = "X: " + X
    moldar.txtPosY.Text = "Y: " + Y
    moldar.txtPosZ.Text = "Z: " + Z
    moldar.txtPosM.Text = "M: " + M + "°"

End Sub

'***** Atualizar valores no PLC *****
Public Sub escreve_geral()
    Dim Errors As Array
    OPCMyGroup_ger.SyncWrite(ITEMMAX_ger, sH_ger, oGeral, Errors)
End Sub
'*****
' *                                     FIM OPC                                     *
'*****

```

Código da função desenhar

```
Private Sub Desenhar()  
    'MOSTRAR DESENHO  
    Dim Desenho As Graphics  
    Desenho = PictureBox1.CreateGraphics  
  
    Dim intX0, intX1, intZ0, intZ1 As Integer  
    Dim intR As Integer  
    Dim intX00, intX11, intZ00, intZ11 As Integer  
    Dim cent(1) As Double  
    Dim xc, zc As Double  
    Dim PenColor As New Pen(Color.Red)  
  
    Desenho.Clear(Color.White)  
  
    For i = 1 To numPts - 1  
  
        '200 é o valor para adaptar o sistema de eixos  
        intX0 = objPontos(i - 1).X  
        intZ0 = objPontos(i - 1).Z  
        intX1 = objPontos(i).X  
        intZ1 = objPontos(i).Z  
        intR = objPontos(i).R  
  
        If intR = 0 Then  
            Desenho.DrawLine(PenColor, intX0 + offsetX, offsetZ - intZ0, intX1 +  
offsetX, offsetZ - intZ1)  
        Else  
            'Desenhar o arco de circunferencia  
            cent = CalculaCentro(intX0, intZ0, intX1, intZ1, intR)  
            xc = cent(0)  
            zc = cent(1)  
  
            For Count = intX0 To intX1 - 2 Step 2  
  
                intX00 = Count  
                intX11 = Count + 2  
                intZ00 = calculaZ(xc, zc, intX00, intR)  
                intZ11 = calculaZ(xc, zc, intX11, intR)  
  
                Desenho.DrawLine(PenColor, intX00 + offsetX, offsetZ - intZ00,  
intX11 + offsetX, offsetZ - intZ11)  
  
                Next Count  
            End If  
        Next i  
    End Sub
```


Código da função determinar centro de circunferência

```
'DETERMINAR O CENTRO DE UMA CIRCUNFERENCIA, TENDO COMO DADOS 2 PONTOS E O RAI0
Public ReadOnly Property CalculaCentro(ByVal x0 As Integer, ByVal z0 As Integer,
ByVal x1 As Integer, ByVal z1 As Integer, ByVal r As Integer) As Array
    Get

        Dim d, a, g, b, x, z As Double
        Dim xc, zc As Double
        Dim centro(1) As Double
        Dim PI As Double

        PI = Math.PI

        d = (Math.Sqrt((x1 - x0) ^ 2 + (z1 - z0) ^ 2)) / 2

        a = Math.Asin(d / Math.Abs(r))

        g = (PI / 2) - Math.Atan(Math.Abs(z1 - z0) / Math.Abs(x1 - x0))

        b = g - a

        x = Math.Cos(b) * r

        z = Math.Sin(b) * r

        If r > 0 Then
            If z0 >= z1 Then
                xc = x1 - x
                zc = z1 - z
            ElseIf z0 < z1 Then
                xc = x0 + x
                zc = z0 - z
            End If
        ElseIf r < 0 Then
            If z0 >= z1 Then
                xc = x0 - x
                zc = z0 - z
            ElseIf z0 < z1 Then
                xc = x1 + x
                zc = z1 - z
            End If
        End If

        centro(0) = xc
        centro(1) = zc

        Return centro
    End Get
End Property
```

Código da função de cálculo da coordenada Z

```
'DETERMINAR A COORDENADA Z EM FUNCAO DE X TENDO COMO DADOS O CENTRO E O RAIOS DA
CIRCUNFERENCIA
Public ReadOnly Property calculaZ(ByVal xc, ByVal zc, ByVal xn, ByVal r) As Double
    Get
        Dim z, zn As Double
        z = (Math.Sqrt(r ^ 2 - (xn - xc) ^ 2))

        If r > 0 Then
            zn = zc + z
        ElseIf r < 0 Then
            zn = zc - z
        End If

        Return zn
    End Get
End Property
```

Código da função de cálculo dos pontos de corte do molde

```
*****
'*          CALCULAR PONTOS DE CORTE DO MOLDE          *
*****
Public Sub CalcularMolde()
    'Determinar os pontos de corte do molde

    Dim intX0, intX1, intZ0, intZ1 As Integer
    Dim intR As Integer
    Dim intX00 As Integer
    Dim cent(1) As Double
    Dim xc, zc As Double
    Dim espessDisco As Integer

    'Inicializa array de pontos criticos
    For i = 0 To larguraMolde
        pontosCriticos(i) = False
    Next i

    For i = 1 To numPts - 1

        intX0 = objPontos(i - 1).X
        intZ0 = objPontos(i - 1).Z
        intX1 = objPontos(i).X
        intZ1 = objPontos(i).Z
        intR = objPontos(i).R

        If intR = 0 Then
            If intX0 = intX1 Then 'recta vertical
                If intZ1 > intZ0 Then
                    alturasZ(intX0) = intZ0
                    pontosCriticos(intX0) = True
                Else
                    alturasZ(intX0) = intZ1
                    pontosCriticos(intX0) = True
                End If
            End If
        End If
    Next i
End Sub
```

```

End If
ElseIf intZ0 = intZ1 Then 'recta horizontal
  'For h = intX0 + 1 To intX1
  For h = intX0 To intX1
    If pontosCriticos(h) = False Then
      alturasZ(h) = intZ0
      If h = (intX0 + espessDisco) Then
        pontosCriticos(h) = True
      Else
        pontosCriticos(h) = False
      End If
    End If
  Next h
Else 'diagonal
  If intZ0 < intZ1 Then
    For d = intX0 + 1 To intX1
      alturasZ(d) = ((intZ1 - intZ0) / (intX1 - intX0)) * (d -
intX0) + intZ0
      pontosCriticos(d) = False
    Next d
  Else
    For d = intX0 To (intX0 + espessDisco)
      If pontosCriticos(d) = False Then
        alturasZ(d) = intZ0
        pontosCriticos(d) = False
      End If
    Next d

    For d = (intX0 + espessDisco) To intX1
      alturasZ(d) = ((intZ1 - intZ0) / (intX1 - intX0)) * ((d -
espessDisco) - intX0) + intZ0
      pontosCriticos(d) = False
    Next d
  End If
End If

Else
  'Pontos dos redondos
  cent = CalculaCentro(intX0, intZ0, intX1, intZ1, intR)
  xc = cent(0)
  zc = cent(1)

  If intR < 0 Then
    For Count = intX0 To (intX0 + espessDisco)
      If pontosCriticos(Count) = False Then
        intX00 = Count
        alturasZ(intX00) = intZ0
        pontosCriticos(intX00) = False
      End If
    Next Count

    For Count = (intX0 + espessDisco) To xc
      intX00 = Count
      alturasZ(intX00) = calculaZ(xc, zc, (intX00 - espessDisco),
intR)
      pontosCriticos(intX00) = False
    Next Count

    For Count = xc To intX1
      If pontosCriticos(Count) = False Then
        intX00 = Count

```

```

        alturasZ(intX00) = calculaZ(xc, zc, intX00, intR)
        pontosCriticos(intX00) = False
    End If
Next Count
End If

If intR > 0 Then
    For Count = intX0 To xc
        If pontosCriticos(Count) = False Then
            intX00 = Count
            alturasZ(intX00) = calculaZ(xc, zc, intX00, intR)
            pontosCriticos(intX00) = False
        End If
    Next Count

    For Count = xc To (xc + espessDisco)
        If pontosCriticos(Count) = False Then
            intX00 = Count
            alturasZ(intX00) = calculaZ(xc, zc, xc, intR)
            pontosCriticos(intX00) = False
        End If
    Next Count

    For Count = (xc + espessDisco) To intX1
        intX00 = Count
        alturasZ(intX00) = calculaZ(xc, zc, (intX00 - espessDisco),
intR)
        pontosCriticos(intX00) = False
    Next Count
End If

End If

Next i

End Sub

```